



# TRANSFERT DE FICHIERS EN MODE MESSAGERIE

Florence Rollin

## ► To cite this version:

Florence Rollin. TRANSFERT DE FICHIERS EN MODE MESSAGERIE. Web. Ecole Nationale Supérieure des Mines de Saint-Etienne; Université Jean Monnet - Saint-Etienne, 1986. Français. NNT: . tel-00807456

**HAL Id: tel-00807456**

**<https://theses.hal.science/tel-00807456>**

Submitted on 3 Apr 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 8ID

# THESE DE DOCTORAT

présentée par

**Florence ROLLIN**

*Spécialité :*

*Informatique, Image, Intelligence Artificielle et Algorithmique*

## TRANSFERT DE FICHIERS EN MODE MESSAGERIE

Soutenue à Saint-Etienne le 16 Décembre 1986, devant la commission d'examen :

Président

G. PUJOLLE

Examineurs

C. CARASSO  
C. KINTZIG  
R. OUZILLOU  
P.A. PAYS  
B. PEROCHE



N° d'ordre : 8ID

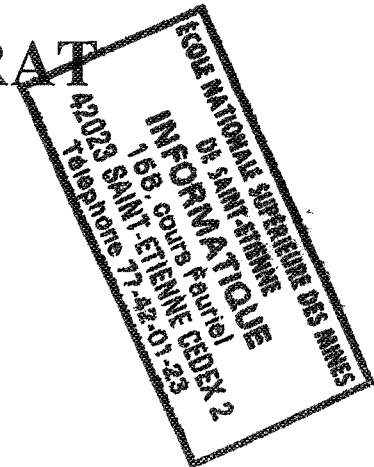
# THESE DE DOCTORAT

présentée par

**Florence ROLLIN**

*Spécialité :*

*Informatique, Image, Intelligence Artificielle et Algorithmique*



## TRANSFERT DE FICHIERS EN MODE MESSAGERIE

Soutenue à Saint-Etienne le 16 Décembre 1986, devant la commission d'examen :

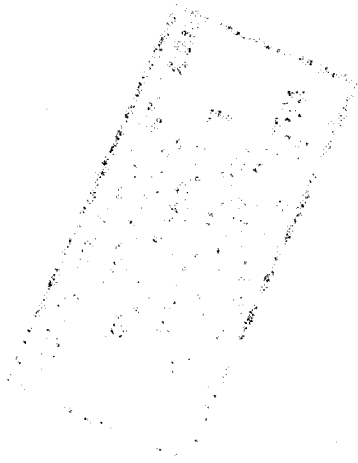
Président

G. PUJOLLE

Examineurs

C. CARASSO  
C. KINTZIG  
R. OUZILLOU  
P.A. PAYS  
B. PEROCHE

PL2318.1.1



# ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT ETIENNE

Directeur	:	M. M. MERMET
Directeur des Etudes et de la formation	:	M. J. CHEVALIER
Directeur des Recherches	:	M. MUDRY
Secrétaire Général	:	Melle M. CLERGUE

## PROFESSEURS DE 1ère CATEGORIE

MM. COINDE	Alexandre	Gestion
FORMERY	Philippe	Mathématiques Appliquées
GOUX	Claude	Métallurgie
LOWYS	Jean-Pierre	Physique
MATHON	Albert	Gestion
PERRIN	Michel	Géologie
PEROCHE	Bernard	Informatique
RIEU	Jean	Mécanique - Résistance des Matériaux
SOUSTELLE	Michel	Chimie
VERCHERY	Georges	Matériaux

## PROFESSEURS DE 2ème CATEGORIE

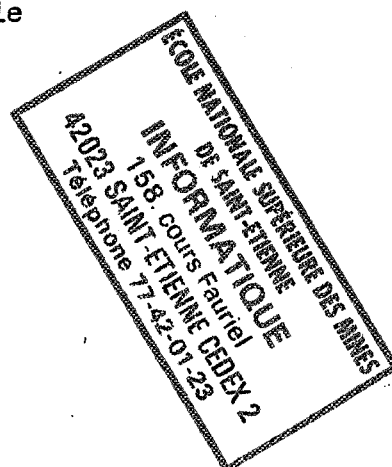
MM. LADET	Pierre	Entreprise et Travaux
PLA	Jean Marie	Mathématiques
TOUCHARD	Bernard	Physique Industrielle

## DIRECTEUR DE RECHERCHE

M. LESBATS	Pierre	Métallurgie
------------	--------	-------------

## MAITRES DE RECHERCHE

MM. BISCONDI	Michel	Métallurgie
CONRAD	Francis	Informatique
DAVOINE	Philippe	Géologie
DRIVER	Julian	Matériaux
Mlle FOURDEUX	Angeline	Métallurgie
MM. GIRARDOT	Jean Jacques	Informatique
GUILHOT	Bernard	Chimie
KOBYLANSKI	André	Métallurgie
LALAUZE	René	Chimie
LANCELOT	Francis	Chimie
LE COZE	Jean	Matériaux
MONTHEILLET	Franck	Matériaux
THEVENOT	François	Chimie
TRAN MINH	Canh	Chimie

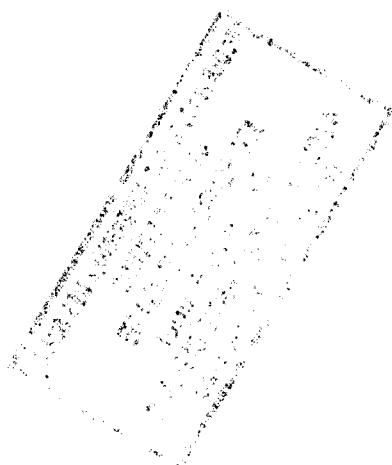


## PERSONNALITES HABILITEES A DIRIGER LES TRAVAUX DE RECHERCHE

MM. CURNIL	Michel	Chimie
MAGNIN	Thierry	Matériaux
THOMAS	Gérard	Chimie

## PROFESSEUR A L'U.E.R. DE SCIENCES DE SAINT ETIENNE

M. VERGNAUD	Jean Marie	Chimie des Matériaux
-------------	------------	----------------------



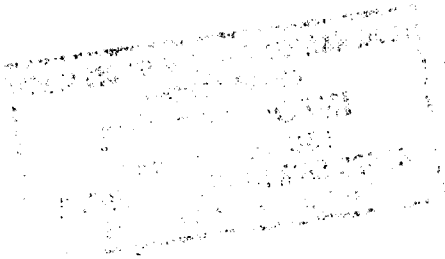
J'exprime ma reconnaissance à :

- Monsieur Guy Pujolle, Professeur à l'Université Pierre et Marie Curie, qui me fait l'honneur de présider le jury de cette thèse ;
- Monsieur Claude Kintzig, Ingénieur au Centre National d'Etudes des Télécommunications, qui est à l'origine de ce travail et qui l'a suivi avec attention jusqu'à la fin ;
- Monsieur Bernard Peroche, Directeur du Département Informatique de l'Ecole des Mines de Saint-Etienne, qui s'est intéressé à ce travail et qui m'a aidé à constituer ce jury ;
- Messieurs Claude Carasso et René Ouzilou, Professeurs à l'Université de Saint-Etienne, qui ont accepté de participer à ce jury ;
- Monsieur Paul-André Pays, Ingénieur à l'Ecole des Mines de Saint-Etienne, qui a guidé mon travail tout au long de ces trois années et qui a supporté, avec patience et abnégation, mes sautes d'humeur.

Je tiens également à remercier :

- Monsieur Patrice Fauvel qui a largement participé à la réalisation de ce projet ;
- Mesdames Mahy, Tocaven et Grün ainsi que Messieurs Pontacq, Soulier et de Baillencourt qui ont, par leur expérience, contribué à l'élaboration de ce projet ;
- Messieurs Philippe Brun et You Yi Zhi, pour leur aide efficace ;
- Madame Geneviève Lallich et Monsieur Segal Sako, pour leurs encouragements ;
- Messieurs Loubet, Darles et Velay, pour avoir assuré avec gentillesse la réalisation de cet ouvrage.





# SOMMAIRE

---

## INTRODUCTION

---

## CHAPITRE 1 : OBJECTIFS, PROBLEMES ET CHOIX

INTRODUCTION	5
LES TRANSFERTS DE FICHIERS EXISTANTS	5
Outils élémentaires	5
UUCP	5
Transferts de fichiers constructeurs	6
Systèmes de fichiers distribués	6
File Transfert Access and Management (FTAM)	6
NOTION DE FICHIER	8
Description d'un fichier	8
NOTION D'UTILISATEUR	11
Gestion des utilisateurs	11
Utilisateur externe ou distant	11
OBJECTIFS	12
Transfert des fichiers de nature universellement connue	12
Cas particulier	13
Transférer la copie d'un fichier	13
Assurer la sécurité	13
S'adapter aux systèmes existants	13
Utilisations envisagées	14
PROBLEMES ET CHOIX DE BASE	15
Comment assurer la sécurité?	15
Comment désigner les utilisateurs distants ?	15
Comment désigner les fichiers distants ?	15
Quel système de communication utiliser ?	16
Résumé des objectifs et choix de base	16



---

## **CHAPITRE 2 : MODELE DE REFERENCE OSI ET MESSAGERIE X400**

<b>MODELE DE REFERENCE OSI</b>	<b>17</b>
Introduction	17
Présentation	17
<b>MESSAGERIE X400</b>	<b>18</b>
Introduction	18
Modèle de système de messages (MHS)	19
Représentation en couches du modèle MHS	21
Syntaxe X409	21

---

## **CHAPITRE 3 : CONCEPTS ET ARCHITECTURE**

<b>INTRODUCTION</b>	<b>23</b>
<b>MODELE DE SYSTEME DE FICHIERS UNIFORMISE (UFS)</b>	<b>23</b>
Description	23
Arborescence et désignation	23
Attribut nature de fichier	24
Autres attributs	24
Correspondance UFS-SGF	25
<b>OPERATIONS ET RESULTATS D'OPERATION</b>	<b>26</b>
Opérations	26
Envoi de fichier	26
Prise de fichier	26
Faisabilité d'envoi	27
Résultats d'opération	28
Résultat d'envoi de fichier	28
Résultat de prise de fichier	28
Résultat de faisabilité d'envoi	29
<b>AUTORISATIONS</b>	<b>30</b>
Association Opération-Autorisation	30
Choix	30
Description	30
Contrôle	32
<b>SIGNALISATION</b>	<b>33</b>
Fonctionnalités	33
Description	33
<b>ARCHITECTURE</b>	<b>34</b>
Modèle fonctionnel	34
Modèle en couches	35
Nom d'O/R	36



---

## **CHAPITRE 4 : COUCHE AGENT UTILISATEUR DE TRANSFERT DE FICHIERS**

<b>INTRODUCTION</b>	<b>37</b>
Méthode de description des services	37
Méthode de description de l'interface supérieure	37
Méthode de description du protocole	39
Méthode de description des actions	39
<b>SERVICES FTUAL</b>	<b>41</b>
Rôle de certains éléments de service	41
Commentaire	41
Priorité de transfert	41
Référence utilisateur	41
Identificateurs de demande et de signalisation	41
Dates associées au fichier copié	42
Possibilités d'installation	42
Renseignements d'installation	42
Résultat anormal	42
Dates associées à une opération	42
<b>INTERFACE UTILISATEUR</b>	<b>44</b>
<b>PROTOCOLE P10</b>	<b>46</b>
Type de Données Lancement	46
Type de Données Réponse	46
Type de Données Problème	47
Rôle de certains Types de Données	47
Type de Données Fichier	47
Type de Données Chaîne-TFMM	48
Types de Données UFS	48
<b>ACTIONS FTUA</b>	<b>49</b>
Actions liées à un envoi de fichier avec demande de résultat	49
Exemple	54

---

## **CHAPITRE 5 : PRESENTATION DE LA REALISATION**

<b>CADRE COSAC</b>	<b>57</b>
<b>SPECIFICATIONS DE REALISATION</b>	<b>57</b>
Objets principaux	57
Interface MTA	58
Les modules principaux	58
Le module DO	58
Le module GTFGM	59
Le module CB	60



Le module MJA	61
Module de gestion	62
Fonctionnalités dépendantes du système	62
Désignations SGF-UFS	63
Nature de fichier	63
Les droits locaux	63

<b>REALISATION</b>	<b>63</b>
--------------------	-----------

---

## **CONCLUSION**

---

## **REFERENCES**

---

## **ANNEXE A**

---

## **ANNEXE B**

---





## Introduction

Le développement rapide des réseaux et de la télématique a mis en évidence la nécessité de définir des normes tant au niveau du matériel que du logiciel. D'importants travaux sont menés par les organismes nationaux et internationaux de normalisation : CCITT, ISO, AFNOR. Ces derniers sont aujourd'hui soutenus par l'ensemble des constructeurs et utilisateurs du monde de l'informatique et des télécommunications.

Le modèle de référence pour l'interconnexion des systèmes ouverts (OSI) est le point de départ d'un ensemble de normes qui définissent le fonctionnement d'un réseau composé de systèmes informatiques hétérogènes répartis à travers le monde. La messagerie X400, appelée également courrier électronique, est sans doute une des premières applications normalisées s'appuyant sur le modèle OSI. De nombreuses autres applications bureautiques et télématiques sont en cours d'étude. En dehors des problèmes politiques et économiques, on peut dire aujourd'hui que la normalisation se porte bien.

C'est dans l'idée de définir de nouvelles applications utilisant les services normalisés dont la messagerie X400 que des groupes de travail Architel ont été créés à l'initiative de M. Claude Kintzig. Les applications actuellement étudiées sont les suivantes :

- TFMM : Transfert de Fichiers en Mode Messagerie.
- STMM: Soumission de Travaux en Mode Messagerie
- CRMM : Conférences Réparties en Mode Messagerie
- BIMM : Bases d'Informations en Mode Messagerie

Tous ces projets ont en commun deux points. Ils utilisent pour communiquer les services de Transfert de Messages X400. Et ils fonctionnent dans un environnement de systèmes et de machines hétérogènes.

Aujourd'hui, le mode de communication en temps réel est utilisé par la plupart des applications téléinformatiques (consultation de bases de données, réservation SNCF...). Cependant, l'utilisation du mode messagerie apparaît intéressant parce qu'il est suffisant dans beaucoup de cas. De plus, il permet d'utiliser les réseaux en dehors des heures de pointe, d'où une diminution des coûts de transfert. En faisant une analogie avec le courrier postal et le téléphone, on peut dire que le mode messagerie et le mode connecté sont complémentaires.

La possibilité offerte par l'OSI de communiquer entre systèmes hétérogènes a mis en évidence la nécessité de définir des applications indépendamment des systèmes d'exploitation sous lesquels elles tournent.

Le système de Transfert de Fichiers en Mode Messagerie (TFMM) permet d'échanger des fichiers entre systèmes hétérogènes. Le système de communication utilisé est le système de Transfert de Messages X400. La première partie de mon travail a été de participer à l'élaboration du service TFMM, à l'intérieur du groupe de travail. La méthode de description utilisée par les recommandations de la série X400 a été reprise pour TFMM. Une première version a été éditée par le CNET en 1985. Une seconde et dernière version va l'être prochainement.

La seconde partie de mon travail a consisté à implanter à la fois sur Unix et sur Multics, en collaboration avec Patrice Fauvel du CNET, une réalisation de TFMM, appelée GTF. GTF s'interface avec le logiciel COSAC-V5 qui réalise l'ensemble des services X400. Une série de documents décrit les spécifications de réalisation. GTF a été écrit en Pascal.

Cette thèse est l'aboutissement de ce travail. Le chapitre 1 présente les hypothèses de départ qui ont orienté notre travail. A savoir :

- traitement des fichiers de *nature universellement connue*. On entend par fichiers de nature universellement connue, des fichiers définis et utilisés dans de nombreux systèmes, mais dont la réalisation peut être différente d'un système à l'autre.
- échange de fichiers, dans leur totalité, et entre des utilisateurs (personnes) se trouvant sur des systèmes connectés par un système de communication MTS (X400).

Dans le chapitre 2, nous présentons le modèle OSI et le système X400 qui constituent le cadre de TFMM.

Nous décrivons, dans le chapitre 3, les différents concepts que nous avons introduits. A savoir :

- *le système de fichier uniformisé* qui est la vision commune qu'a TFMM, de tous les systèmes de gestion de fichiers.
- les *opérations* et les *autorisations* qui sont à la base du fonctionnement de TFMM.

Nous terminons ce chapitre, en donnant l'architecture du système TFMM.

Nous donnons, dans le chapitre 4, une description rapide du fonctionnement du système TFMM. Pour avoir une description complète, on se reportera à l'annexe A.

Dans le chapitre 5, nous donnons un aperçu de la réalisation GTF. Pour plus de détails, on se reportera à l'annexe B.

En conclusion, nous effectuons le bilan de ce travail.



# **Chapitre 1**

## **OBJECTIFS, PROBLEMES ET CHOIX**

### **1 INTRODUCTION**

Notre but est de définir un transfert de fichiers fonctionnant dans un cadre hétérogène, qui soit simple mais qui prenne en compte les besoins les plus fréquents des utilisateurs.

Après avoir recensé les transferts de fichiers existants et présenté les notions de fichier et d'utilisateur, nous énonçons nos objectifs. Nous exposons ensuite les problèmes qui se sont posés et finalement les choix que nous avons faits.

### **2 LES TRANSFERTS DE FICHIERS EXISTANTS**

#### **2.1 OUTILS ELEMENTAIRES**

Le logiciel Kermit permet d'effectuer depuis une première machine qui joue le rôle d'un terminal, une session sur une seconde machine. Il permet également de transférer des fichiers entre les 2 machines en lançant un second Kermit sur la machine hôte. Durant la session et lors des transferts de fichiers, Kermit assure les transformations entre les deux types de codage de caractères. Différents mécanismes (contrôle de flux et d'erreurs) sont mis en place pour assurer le transfert fiable des données. Prévu pour fonctionner sur ligne asynchrone simple entre un micro-ordinateur et une autre machine, il est également utilisé sur des réseaux via un PAD et entre toutes sortes de machines. Du fait de sa gratuité, c'est à l'heure actuelle un des outils les plus répandus pour transférer des fichiers (textes) entre des systèmes hétérogènes.

Le logiciel FTP, utilisé dans le réseau ARPA, fonctionne comme Kermit par émulation de terminal. Il définit, en plus, un mini langage de commande indépendant des systèmes sur lesquels il tourne.

#### **2.2 UUCP**

Le logiciel UUCP permet de transférer des fichiers entre machines Unix. Les fichiers Unix ayant pour seule structure l'octet, toutes les informations contenues dans le fichier sont conservées pendant le transfert. UUCP fonctionne en "batch" et en mode "store and forward". Lorsqu'un utilisateur demande la copie et l'envoi d'un fichier vers une machine distante, le fichier est mis dans une file d'attente. Il

sera traité à la prochaine activation d'UUCP. Le fichier peut être transporté sur plusieurs machines relais avant d'atteindre sa destination. UUCP gère, de façon rudimentaire, des droits pour les utilisateurs distants.

Le système de communication utilisé est une ligne asynchrone simple ou son émulation. UUCP ne fonctionne donc pas sur des systèmes de communication normalisés.

### 2.3 TRANSFERTS DE FICHIERS CONSTRUCTEURS

Les constructeurs informatiques commercialisent dans le cadre de leur architecture de réseaux (DSA pour Bull, SNA pour IBM...) des outils pour transférer des fichiers entre des machines de la même famille. Les différentes organisations et méthodes d'accès (séquentiel, séquentiel indexé...) sont conservées. D'autres sociétés offrent des transferts de fichiers compatibles avec ceux des plus grands constructeurs.

Ces logiciels fonctionnent le plus souvent en mode connecté en utilisant les moyens de communication offerts par l'architecture de réseaux.

### 2.4 SYSTEMES DE FICHIERS DISTRIBUES

Prévu pour fonctionner sur un réseau local à haut débit, un système de fichiers distribués offre un accès banalisé et transparent à tous les fichiers se trouvant sur l'ensemble des machines connectées au réseau. Ce système fonctionne sur un modèle "Client-Serveur". Le "Serveur" se trouve sur le même site que le fichier qui est accédé, tandis que le "Client" utilisant le fichier se trouve sur une machine distante. Ils communiquent en mode connecté. Un serveur de noms permet de désigner et de situer n'importe quel fichier.

Parmi les produits les plus récents, les systèmes Network File System (NFS) de Sun Microsystems et Remote File Sharing (RFS) d'AT&T sont en concurrence. Sorti le premier, NFS tourne sur différents systèmes d'exploitation dont Unix, VMS et MSDOS. RFS s'applique spécifiquement à Unix. Ces produits sont très utiles car ils permettent une banalisation de l'ensemble des ressources d'un réseau local.

### 2.5 FILE TRANSFERT ACCESS AND MANAGEMENT (FTAM)

FTAM [11] est un service de gestion de fichiers qui se situe dans la couche Application du modèle de référence OSI. Il est en cours de normalisation à l'ISO. Il offre à un utilisateur (humain ou application), dans le cadre de systèmes ouverts, les moyens de gérer un fichier, d'y accéder et de le transférer.

FTAM définit un modèle commun de système de fichiers virtuels appelé Virtual Filestore. Ce dernier est défini par des attributs, des structures et des opérations. Les attributs reflètent la gestion du fichier (stockage, sécurité, nom, date). Les

structures donnent l'organisation des données contenues dans le fichier. Ces données font l'objet d'accès et de transfert. Les opérations sont soit des opérations sur le fichier complet (création, ouverture, modification d'un attribut), soit des opérations sur les données contenues dans le fichier (lecture, remplacement, insertion d'articles).

Le modèle de fonctionnement est également basé sur un modèle "Client-Serveur". Les deux entités en présence sont appelées "File Service Initiateur" pour le client et "File Service Respondeur" pour le serveur. Le service est orienté connexion. Une session est divisée en plusieurs phases : reconnaissance des interlocuteurs, reconnaissance du fichier à traiter, reconnaissance des informations à manipuler et enfin, traitement proprement dit.

Ayant pour ambition de prendre en compte le plus de systèmes possibles, le service FTAM est devenu assez complexe. Sa normalisation a d'ailleurs pris du retard. Cette complexité entraîne des coûts et des difficultés de mise en oeuvre importants pour les services FTAM. De plus, pour que l'écriture ou la réécriture d'une application utilisant les services FTAM soit rentable, il faut que celle-ci soit assez stable et très souvent utilisée.



### 3 NOTION DE FICHIER

Qu'est-ce qu'un fichier ? C'est un ensemble d'informations, formant un tout, défini et utilisé dans le cadre d'une application, mais ayant généralement une durée de vie supérieure aux processus qui le manipulent.

Les systèmes de gestion de fichiers (SGF) offrent une méthode pour organiser l'ensemble des fichiers du système (par exemple en arborescence), et un moyen pour désigner un fichier parmi cet ensemble. A ce niveau, un fichier est vu comme un contenant. Le SGF lui associe un certain nombre d'attributs, tels que des noms, des dates (création, dernier accès, dernière mise à jour), des relations avec des utilisateurs, etc.

Les SGF offrent également des organisations pour structurer le contenu des fichiers en articles et des méthodes d'accès à ces articles. Ces mécanismes prédéfinis déchargent les programmes d'application d'une partie de la gestion des fichiers. Par exemple :

Organisation	Méthodes d'accès
séquentiel	séquentiel
relatif	séquentiel ou aléatoire
séquentiel indexé	séquentiel ou direct par clé

On retrouve ces types d'organisation et de méthodes d'accès, dans la plupart des systèmes, même s'ils diffèrent dans leur réalisation. Certains systèmes, comme Unix, ne gèrent pas ces mécanismes au niveau du SGF. Le SGF d'Unix considère les fichiers comme des suites d'octets. Les organisations de fichier et fonctions d'accès sont offertes sous forme de bibliothèques de fonctions à intégrer dans les programmes d'application.

Ce que l'on vient de décrire c'est l'aspect visible du SGF, c'est-à-dire ce que les utilisateurs et les programmes d'application voient. L'autre aspect du rôle d'un SGF consiste à faire correspondre à cette vision une représentation sur mémoire secondaire.

Mais la façon dont le SGF gère un fichier, ne suffit pas à le décrire complètement. Il faut le replacer dans l'application qui le définit et l'utilise.

#### 3.1 DESCRIPTION D'UN FICHIER

On entend par application, l'ensemble constitué de plusieurs programmes et fichiers de données qui fonctionnent dans un environnement incluant du matériels et des intervenants humains. Le tout réalise une fonctionnalité précise. Donnons quelques exemples d'applications.

- Soit une application bancaire dont la fonctionnalité est de gérer les comptes des clients d'une banque. Elle est constituée d'un ou plusieurs fichiers de données, de programmes effectuant la mise à jour des données, et de programmes de statistiques, d'un système informatique (ordinateur, terminal) et d'un employé de la banque.
- Soit une application de traitement de texte, dont la fonctionnalité est la saisie et l'impression de texte. Elle est constituée de fichiers de textes, de programmes de saisie et d'impression, d'un système informatique (ordinateur, terminal, imprimante), et d'un utilisateur.

La nature d'un fichier, c'est à dire la composition et la signification des informations qu'il contient, est déterminée par le rôle qu'il joue dans l'application.

- Exemple 1 : Un fichier de données de l'application bancaire est composée d'éléments "client". Chaque élément "client" est composée d'éléments "numéro de compte", "nom du client", "solde du compte", etc. Un élément "numéro de compte" est composé de 10 éléments "chiffre". Et ainsi de suite...
- Exemple 2 : Un fichier texte de l'application traitement de textes est composé d'éléments "titre", "auteur", "texte". Un élément "titre" est composé d'une suite d'éléments "mot". Un élément "mot" est composé d'une suite d'éléments "caractère". Un élément "texte" est composé d'une suite d'éléments "paragraphe". Et ainsi de suite...

Les programmes de l'application effectuent des opérations sur les éléments composant le fichier.

- Exemple 1 : Ajout d'un client. Créditer de telle somme le compte de tel client. Imprimer les numéros de compte dont la somme est négative.
- Exemple 2 : Détruire une phrase. Mettre en majuscules un mot.

Une nature de fichier représente la sémantique d'un fichier, indépendamment de sa réalisation.

La réalisation d'une application fait correspondre à la nature du fichier, une **structuration**, en effectuant des choix parmi les possibilités offertes par le SGF (organisations) et par le langage de programmation utilisé (types de données). De même, les opérations sur les éléments correspondent dans la réalisation à des fonctions ou à des procédures agissant sur la structure du fichier.

- Exemple 1 : Supposons que l'application bancaire soit écrite en Cobol dans un système IBM. Le fichier est un séquentiel indexé. A chaque article, correspond un élément "client". L'accès à un article se fait par clé sur le numéro de compte. Chaque opération est réalisée par une fonction Cobol qui utilise les méthodes d'accès.
- Exemple 2 : L'application est écrite en C sur un système Unix. Dans un fichier, on trouve d'abord d'une structure C qui contient les informations : auteur, titre, etc, puis une suite de lignes de longueur indéfinie.

La réalisation d'une application, et en particulier la structuration du fichier, sera donc différente d'un système à l'autre. Sur un système donné, une même application peut donner lieu à plusieurs réalisations différentes.

La représentation d'une application (programmes exécutables et fichiers sur disque) dépend fortement du système et de la machine. La représentation du fichier sur disque dépend de la façon dont sont codés les organisations (séquentiel, relatif, etc) et les types de données (entier, record, tableau, etc).

Pour récapituler, on peut dire qu'un fichier est défini à trois niveaux.

- Par sa nature : la signification que lui donne l'application.
- Par sa réalisation : la structuration que lui donnent les programmes de l'application.
- Par sa représentation : le codage défini par le système.

## **4 NOTION D'UTILISATEUR**

On peut constater que la plupart des systèmes d'exploitation gèrent la notion d'utilisateur. Par contre, les notions de systèmes distants et d'utilisateurs distants apportées par la connexion à un système de communication ne sont généralement pas intégrées au système d'exploitation.

### **4.1 GESTION DES UTILISATEURS**

Seuls les utilisateurs enregistrés ont accès au système. Les systèmes d'exploitation associent aux utilisateurs un certain nombre d'attributs parmi lesquels :

- des attributs d'identité (nom de login, nom de groupe, alias) ;
- des attributs de droits (droits sur des fichiers, droits d'exécuter des programmes, priorité d'exécution, maximum d'espace disque ou mémoire allouable, etc) ;
- des attributs liés à la comptabilité (temps machine utilisé, espace disque utilisé, etc).

Les mécanismes de protection et de comptabilité sont basés sur la notion d'utilisateur. Ils consistent à définir et à contrôler les droits et les devoirs de chaque utilisateur enregistré.

Dans les relations entre utilisateurs et fichiers, interviennent d'une part les droits d'accès, d'autre part la gestion de l'espace disque alloué.

### **4.2 UTILISATEUR EXTERNE OU DISTANT**

On entend par utilisateur externe ou distant (par opposition à utilisateur enregistré ou local), un utilisateur se trouvant sur un système distant relié par un système de communication.

Sauf dans le cas d'un système distribué (où tous les utilisateurs ont accès à toutes les ressources du système), un utilisateur externe ou distant n'a aucun droit dans le système.

Par contre, un utilisateur externe ou distant peut apparaître dans les applications de communication. Il peut, par exemple, être l'expéditeur ou le destinataire de messages.

## 5 OBJECTIFS

### 5.1 TRANSFERT DES FICHIERS DE NATURE UNIVERSELLEMENT CONNUE

On a vu dans la section 3, qu'un fichier est défini à trois niveaux : nature, réalisation et représentation. Il est impensable de transférer entre systèmes hétérogènes, le fichier selon le niveau le plus bas qui est fortement dépendant du système.

La première approche possible consiste à transférer les fichiers suivant leur réalisation. On définit pour cela des organisations, des méthodes d'accès et des types de données standard. Sur chaque système, on réalise ces définitions standard. C'est l'approche de FTAM.

Notre approche consiste à transférer les fichiers selon leur nature. Il est, pour le moment, impossible de décrire la nature de n'importe quel fichier appartenant à une application. Par contre, on peut constater qu'il existe des natures de fichier déjà définies, et largement utilisées. C'est le cas des fichiers de texte, des fichiers de sources de programmes, des documents normalisés.

Exemples : La nature de fichier texte définit un fichier texte, comme étant une suite d'éléments "lignes". Un élément "ligne" est une suite d'éléments "caractères". Un élément "caractère" appartient à l'alphabet normalisé IA5. La nature de fichier source Pascal, est définie par la syntaxe et la sémantique du langage Pascal. La nature de document ODA [14] définit les notions de document, de paragraphe et de texte.

On trouve sur de nombreux sites, des programmes d'application utilisant les fichiers ainsi définis.

Exemples : Les éditeurs, ligne ou pleine-page, utilisent la nature de fichier texte. Les compilateurs Pascal utilisent la nature de fichier source Pascal. Et bientôt, on trouvera des logiciels utilisant les natures de fichier document ODA [14].

Nous offrons, dans le système TFMM, la possibilité de transférer les fichiers de nature universellement connue. Avec le développement de la bureautique et de la télématique, de nouvelles natures de fichier vont être définies. Le système TFMM sera capable de les intégrer.

Le système TFMM définit pour chaque nature de fichier, une syntaxe de transfert. Sur chaque site, TFMM connaît la représentation que donne le système à chaque nature de fichier. Son travail consiste à transférer, la représentation locale d'un fichier, dans la syntaxe de transfert correspondant à sa nature. Puisque l'information "nature de fichier" ne se trouve pas dans le système de gestion de fichiers, c'est l'utilisateur qui doit la fournir au système TFMM.

### 5.1.1 Cas particulier

On voudrait pouvoir également transférer des fichiers qui n'ont pas une nature universellement connue. On définit pour cela une nature particulière de fichier, dite indéterminée. La syntaxe de transfert, associée à la nature indéterminée, est une suite d'octets. Sur chaque site, TFMM associe à la nature indéterminée une représentation locale particulière.

On peut utiliser la nature indéterminée pour transférer des fichiers codés selon une syntaxe comprise par l'expéditeur et par le destinataire du fichier. On peut également l'utiliser pour transférer des fichiers quelconques entre sites homogènes.

## 5.2 TRANSFERER LA COPIE D'UN FICHIER

Dans le système TFMM, les fichiers sont transférés dans leur entier. En cela, TFMM se démarque des systèmes qui permettent la création, la modification et la destruction de fichier à distance. Le principe est le suivant :

- Prise de copie d'un fichier sur un site de départ.
- Transfert de la copie vers un ou plusieurs sites d'arrivée.
- Sur chaque site d'arrivée, création d'un nouveau fichier contenant la copie.

Le système TFMM n'offre pas à des applications des outils pour définir et utiliser des fichiers. Mais il permet de transporter des fichiers entre des sites où se trouvent des applications qui les utilisent.

## 5.3 ASSURER LA SECURITE

On veut également que le système TFMM assure la sécurité. C'est l'un des points cruciaux de toute application de communication. En effet, il ne faut pas qu'il permette de passer outre les différents contrôles mis en place par les systèmes d'exploitation et mette ainsi en péril l'intégrité des systèmes.

Il faut éviter, dans TFMM, qu'un utilisateur externe ait accès aux fichiers d'un site sans aucun contrôle. Il faut également éviter qu'un utilisateur enregistré se serve de TFMM pour accéder à des fichiers sur lesquels il n'a aucun droit.

## 5.4 S'ADAPTER AUX SYSTEMES EXISTANTS

Il ne s'agit pas dans TFMM de définir un système d'exploitation et de gestion de fichiers distribué et hétérogène. Mais il s'agit de permettre, quand c'est possible, le transfert de fichiers entre systèmes d'exploitation classiques. C'est moins ambitieux, mais tout aussi utile.

TFMM doit donc être capable de s'adapter à différents systèmes. Pour cela, il reprend un ensemble de notions communes à tous les systèmes, tout en respectant les caractéristiques de chacun. Il définit, également, de nouveaux mécanismes qui complètent ceux des systèmes d'exploitation, en particulier pour la sécurité.

### 5.5 UTILISATIONS ENVISAGEES

Le système TFMM pourrait être utilisé par un groupe d'utilisateurs dispersés géographiquement, et travaillant à la réalisation d'un projet commun (par exemple : réalisation d'un logiciel). Ils pourraient ainsi s'échanger des fichiers (manuel de référence, spécifications de réalisation, sources). Le système TFMM pourrait également être utilisé par des sociétés ou des organismes (banques, administrations, sociétés informatiques) pour envoyer à leurs clients des informations (catalogue produits, liste des points de vente, bulletin interne d'informations). Un administrateur réseau pourrait, par exemple, utiliser le système TFMM, pour envoyer la nouvelle liste des sites connectés.

En bref, tous ceux qui ont besoin de transférer, de façon ponctuelle ou régulière, des fichiers entre des systèmes distants hétérogènes peuvent utiliser TFMM.

## 6 PROBLEMES ET CHOIX DE BASE

### 6.1 COMMENT ASSURER LA SECURITE?

On a vu précédemment, que les systèmes classiques ne gèrent pas la notion d'utilisateur externe ou distant. Etant donné que nous avons pour objectif de nous adapter aux systèmes existants, le principe de fonctionnement suivant est adopté :

- Les opérations d'échange de copies de fichiers offertes par TFMM ont toujours lieu entre des utilisateurs, connus sur chacun des sites concernés. On définit ainsi la notion d'**utilisateurs responsables**. Dans une opération, on a toujours un utilisateur responsable sur le site de départ et un (des) utilisateur(s) responsable(s) sur le(s) site(s) d'arrivée.
- Le contrôle est effectué par le système TFMM de deux façons ;
  - d'une part, en obtenant l'accord des différents utilisateurs responsables ;
  - d'autre part, en maintenant les droits de chacun d'eux, sur les sites où ils se trouvent. L'**utilisateur responsable sur le site de départ** doit posséder des **droits de lecture** sur le fichier copié. Chacun des **utilisateurs responsables sur le site d'arrivée**, devient le **propriétaire** du fichier créé, il doit donc posséder les **droits de création**.

### 6.2 COMMENT DESIGNER LES UTILISATEURS DISTANTS ?

On a besoin de désigner les utilisateurs distants intervenant dans le transfert. Il s'agit premièrement de désigner un site (machine, système), deuxièmement de désigner un utilisateur sur ce site. On pourrait, pour l'utilisateur, utiliser le nom de login, et pour le site, une méthode d'adressage offerte par le système de communication employé.

Il apparaît plus intéressant d'utiliser une **désignation logique d'utilisateur**, indépendante des différents systèmes d'exploitation et de communication et de faire correspondre cette désignation logique avec un utilisateur réel.

### 6.3 COMMENT DESIGNER LES FICHIERS DISTANTS ?

Un autre des problèmes posés par TFMM, est la méthode de désignation des fichiers distants. En effet, chaque système d'exploitation utilise des méthodes d'identification différentes.

La solution la plus générale est de définir une désignation indépendante des systèmes d'exploitation, appelée **désignation universelle de fichiers** et de faire correspondre cette désignation avec les noms réels des fichiers.



#### 6.4 QUEL SYSTEME DE COMMUNICATION UTILISER ?

Le choix du système de communication doit prendre en compte les objectifs et les principes généraux qui viennent d'être présentés.

On peut constater qu'un fichier est transféré en entier et qu'il n'existe aucun lien entre plusieurs opérations de transfert. De ce fait, le fonctionnement en temps réel n'apparaît pas primordial. De ces constatations, on déduit que le mode de communication par commutation de messages est adapté à notre application. Ce mode permet une meilleure utilisation du réseau ; d'où des coûts moindres de transfert ce qui n'est pas négligeable si les fichiers sont gros.

Le système TFMM utilise le système de communication MTS défini dans la messagerie X400 (cf Chapitre 2, section 2). Le mode d'adressage des utilisateurs par nom d'O/R, résout notre problème de désignation des utilisateurs.

Les avantages que nous avons à utiliser le système MTS, sont les suivants :

- son existence : on se décharge ainsi de l'aspect communication. Dans un proche avenir, des réalisations de la messagerie X400 (donc de MTS) seront disponibles sur de nombreux systèmes.
- sa normalisation : il est inutile de revenir sur les avantages de la normalisation, chacun connaissant les inconvénients de l'incompatibilité.

#### 6.5 RESUME DES OBJECTIFS ET CHOIX DE BASE

En résumé, le système TFMM permet le transfert de copies de fichier entre des utilisateurs se trouvant sur des systèmes distants, hétérogènes, connectés par un système de communication MTS. Les fichiers sont transférés dans leur entier, et selon leur nature.

TFMM a été défini pour être utilisable dans de nombreux systèmes d'exploitation existants. Des solutions pour maintenir la sécurité des systèmes ont été trouvées.

## Chapitre 2

# MODELE DE REFERENCE OSI ET MESSAGERIE X400

## 1 MODELE DE REFERENCE OSI

### 1.1 INTRODUCTION

Les réseaux téléinformatiques sont nés dans les années 50. Il s'agissait, au début de déporter quelques terminaux. Au début des années 70, sont apparus les premiers réseaux d'ordinateurs : d'abord sur une échelle limitée, et bientôt sur échelle nationale et internationale, avec l'apparition des réseaux publics de transmission de données. On voit apparaître vers 74, les premières architectures de réseaux chez les constructeurs : SNA chez IBM, puis Decnet chez DEC, DSA chez Bull, et BNA chez Burroughs. Toutes ces architectures sont fondées sur une technique de structuration en couches.

Malheureusement, les architectures de réseaux proposées par les différents constructeurs ne sont généralement pas compatibles entre elles, tant sur le plan de la structuration des fonctions que sur le plan de la définition des protocoles de communication. Cela exclut la possibilité d'interconnecter des équipements informatiques provenant de constructeurs différents. Avec la croissance très rapide du marché des réseaux, le besoin de normes conçues pour des réseaux hétérogènes se fait sentir. Ainsi en 77, l'organisation internationale de normalisation ISO crée un groupe chargé d'étudier des normes d'Interconnexion de Systèmes Ouverts (OSI).

Le premier travail de l'OSI fut d'établir un modèle de référence qui définit une architecture de réseau au sein de laquelle s'implantent les différents protocoles normalisés.

### 1.2 PRESENTATION

Le modèle de référence OSI (CCITT série X200 [1] [2], ISO 7498 [10]) est un modèle abstrait d'architecture de réseaux. L'utilisation d'un modèle abstrait correspond à la nécessité d'indépendance vis à vis des réalisations. Sont définies également, les conditions sous lesquelles un système réel est conforme au modèle. Il est dit alors, système ouvert.

Le modèle OSI donne, d'une part, les principes de structuration en couches. Il définit en particulier les concepts de **couche**, d'**interface**, de **service**, de **protocole**, de **connexion** [2]. Il spécifie, d'autre part, une décomposition en 7 couches, et définit le rôle de chacune d'elles. C'est l'architecture OSI (Figure 2.1).

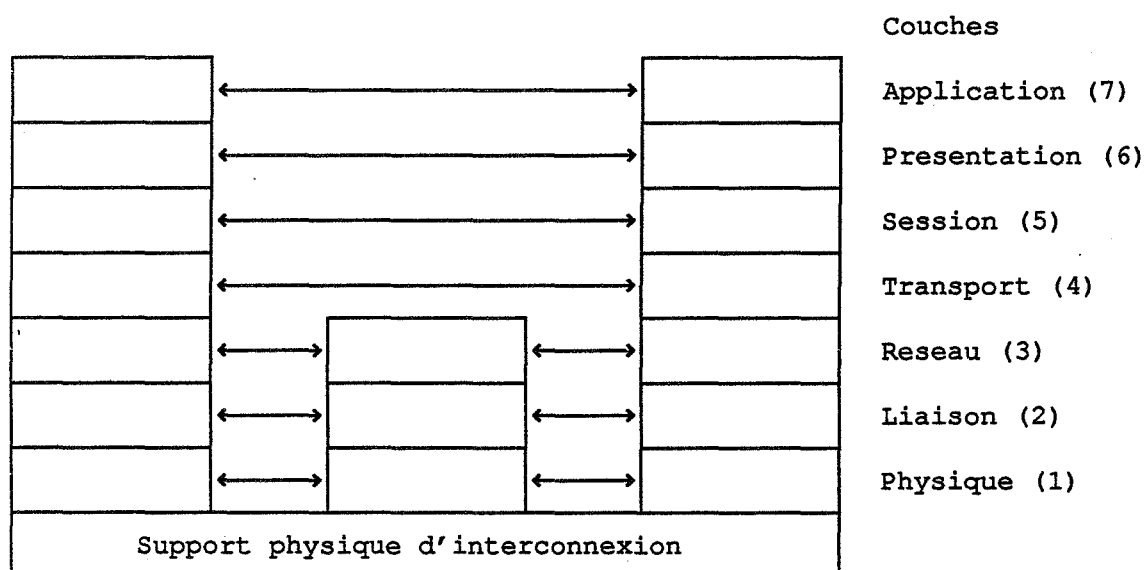


Figure 2.1 : Architecture OSI

Les couches 1 et 2 réalisent les connexions simples entre équipements. Les couches 3 à 5 gèrent un réseau d'ordinateurs répartis géographiquement. Elles offrent aux applications informatiques, télématiques et bureautiques situées dans les couches 6 et 7, les moyens de communiquer.

## 2 MESSAGERIE X400

### 2.1 INTRODUCTION

La messagerie X400 vise à définir un ensemble de normes permettant l'échange international de messages en **mode enregistrement et retransmission**. C'est-à-dire qu'il n'y a pas de connexion entre les correspondants, mais que le message est transféré de proche en proche, depuis le système où se trouve l'expéditeur, vers celui du destinataire. La messagerie X400 est également appelée **courrier électronique**, par analogie au courrier postal.

Deux services de Messagerie ont été définis. Le **service de Messagerie de Personne à Personne** (service d'IPM) [6] permet les communications entre individus. Le **service de Transfert de Messages** (service MT) [5] concerne le service général de transfert de messages.

## 2.2 MODELE DE SYSTEME DE MESSAGES (MHS)

Le modèle de système de messages (MHS) décrit le rôle et le fonctionnement des différentes entités fonctionnelles qui rendent les services de messagerie.

Un utilisateur est un individu ou une application informatique. Il est désigné comme **expéditeur** lorsqu'il envoie un message, comme **destinataire** lorsqu'il en reçoit. Un message est constitué d'une **enveloppe** et d'un **contenu**. L'enveloppe contient différentes informations parmi lesquelles on trouve les noms de l'expéditeur et des destinataires, la date d'émission, une demande d'accusé de réception. Le message peut contenir des données, de la voix, des images. Un message a un ou plusieurs destinataires.

Un expéditeur prépare un message à l'aide de son **Agent d'Utilisateur (UA)**. Ce dernier le dépose auprès d'un **Agent de Transfert de Messages (MTA)**. Les MTA qui constituent le **Système de Transfert de Messages (MTS)**, relaient le message jusqu'aux UA destinataires, qui les remettent à leurs utilisateurs. Les MTA se servent de l'enveloppe pour transférer un message. Ils ne modifient pas le contenu.

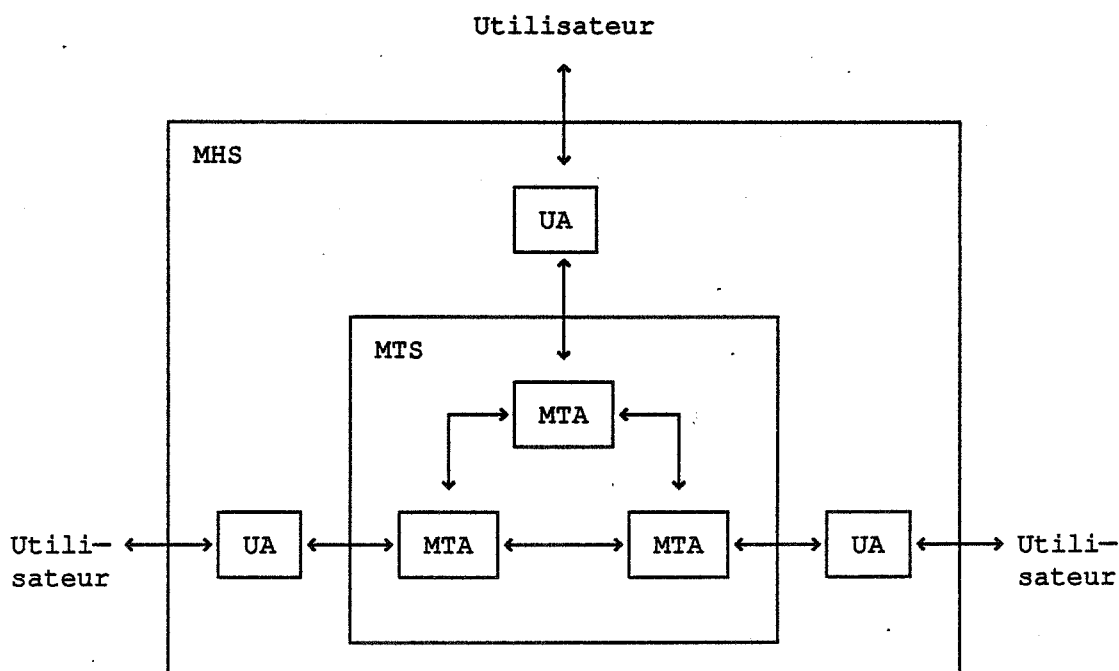


Figure 2.2 : Structure fonctionnelle du modèle MHS

Les UA sont regroupés en classes d'après les types de contenu de messages, qu'ils peuvent traiter. Les UA d'une même classe sont dits UA coopérants. Les UA d'IPM constituent une classe particulière d'UA coopérants.

## Domaine de gestion

Un ensemble formé par au moins un MTA et zéro, un ou plusieurs UA appartenant à un même organisme, constitue un **Domaine de Gestion (MD)** [3]. Un MD, géré par un organisme public (PTT ou équivalent), est appelé **Domaine de Gestion Administratif (ADMD)**. Un MD, géré par un organisme privé, est appelé **Domaine de Gestion Privée (PRMD)**.

Les recommandations CCITT (X400) décrivent le fonctionnement des messageries publiques (ADMD); les avis ISO traitent des messageries privées (PRMD).

## Nom d'O/R

Un **nom d'O/R** [3] désigne un utilisateur dans le système de messagerie. La norme X400 définit un certain nombre d'attributs standard tels que les attributs individuels (nom, prénom), les attributs géographiques (pays), les attributs organisationnels (société, division, poste ou fonction) et des attributs architecturaux (adresse télex, nom d'ADMD, nom de PRMD). Ces attributs sont utilisés pour construire différentes formes de nom d'O/R.

Ces attributs peuvent être classés en deux catégories. Les attributs de base servent à désigner le domaine de gestion (MD) auquel appartient l'utilisateur. Ils sont fixés par les instances internationales. Les attributs "utilisateur" désignent un utilisateur à l'intérieur d'un domaine de gestion. Plusieurs possibilités sont offertes dans la norme. C'est le domaine de gestion qui effectue un choix en fonction des UA qu'il dessert. Par contre pour désigner un UA distant, il doit pouvoir traiter toutes les formes définies par la norme.

Une **adresse d'O/R** permet de situer géographiquement un utilisateur dans le système de messagerie. Pour remettre un message à son destinataire, le système de messagerie doit être capable d'associer un nom d'O/R à une adresse d'O/R. Cela peut être fait par l'intermédiaire d'un service d'annuaire. Comme ce dernier n'est pas encore défini, on utilise actuellement pour désigner les utilisateurs, des noms d'O/R qui sont aussi des adresses d'O/R. Sont actuellement définis quatre types de nom d'O/R [3].

## Configuration physique

Un UA et un MTA peuvent appartenir à un même système (cf Figure 2.3 système S3). Ils sont dits corésidents. Un UA est dit autonome s'il se trouve sur un système distinct (terminal intelligent) (cf Figure 2.3 système S1). Un système peut contenir un MTA et aucun UA (cf Figure 2.3 système S2). On parle alors de MTA autonome.

### 2.3 REPRESENTATION EN COUCHES DU MODELE MHS

Les fonctions de messagerie sont réparties en deux couches dans la couche Application du modèle de référence OSI. La **couche Agent Utilisateur (Couche UAL)** contient les fonctions d'UA. Elle est constituée un ensemble d'entités d'UA (UAE) qui dialoguent à l'aide d'un protocole de nom générique Pc. Le protocole P2 utilisé par les entités d'UAE d'IPM, est une version spécifique de protocole Pc.

La **couche Transfert de Messages (Couche MTL)** fournit les services de Transfert de Messages. Elle est constituée d'entités de MTA (MTAE) et d'entités de Dépôt et de Remise (SDE). Ces dernières gèrent un protocole d'accès à un MTA pour les UA autonomes. Les MTAE dialoguent à l'aide du protocole P1. Les SDE utilisent le protocole P3 pour communiquer avec les MTAE.

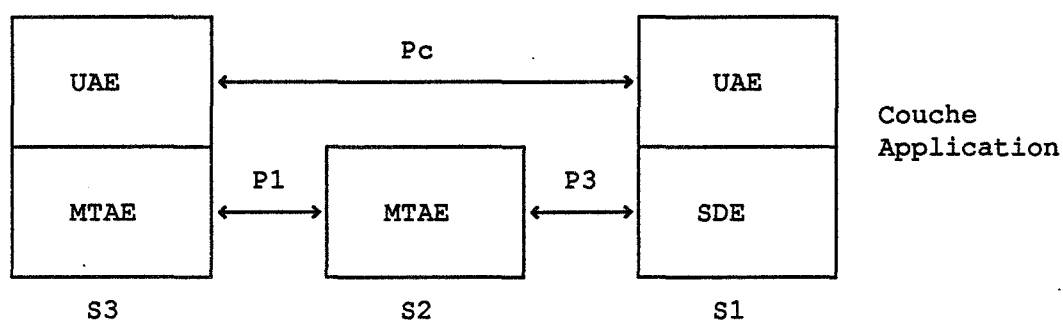


Figure 2.3 : Structuration en couches

### 2.4 SYNTAXE X409

La syntaxe X409 est le moyen utilisé dans les applications CCITT (messagerie et échange de documents), pour décrire et coder les différentes informations constituant les protocoles de communications.

Chaque information a un type et une valeur. Il existe des types prédéfinis primitifs tels que les entiers, les booléens, les caractères et des types prédéfinis constructeurs tels que les choix, les séquences et les ensembles. L'ensemble de ces types permet de construire les types de données que l'on trouve dans les protocoles.

La **notation standard** est l'ensemble des conventions utilisées pour écrire les protocoles dans les normes. C'est un langage décrit à l'aide de la notation BNF.

La **représentation** est l'ensemble des règles de codage employées pour transmettre les données de protocole sous la forme d'une séquence d'octets. C'est un codage de type : Identificateur, Longueur, Valeur (TLV).

La syntaxe X409 a été reprise comme norme ISO sous le nom ASN.1 [12] [13]. Elle est utilisée pour de nombreux protocoles d'applications.

## Chapitre 3

### CONCEPTS ET ARCHITECTURE

#### 1 INTRODUCTION

Un utilisateur dispose d'opérations pour envoyer des fichiers sur des sites distants et pour ramener des fichiers se trouvant sur des sites distants. Il connaît les résultats des opérations qu'il a lancées de manière asynchrone. L'utilisateur qui est à l'origine d'une opération est appelé **utilisateur initiateur**.

La vision qu'ont les utilisateurs des fichiers distants, se fait à travers un modèle virtuel des différents SGF.

On appelle **utilisateurs impliqués**, les utilisateurs concernés par une opération, mais qui n'en sont pas à l'origine. Pour qu'une opération réussisse, il faut avoir l'**autorisation** d'un utilisateur impliqué. Un mécanisme de **signalisation** permet d'informer un utilisateur impliqué de toute opération effectuée par un autre utilisateur sur un fichier dont il s'est porté responsable dans l'autorisation.

Ce chapitre présente les différents concepts et mécanismes utilisés. Il se termine par la présentation du modèle et de l'architecture du service TFMM.

#### 2 MODELE DE SYSTEME DE FICHIERS UNIFORMISE (UFS)

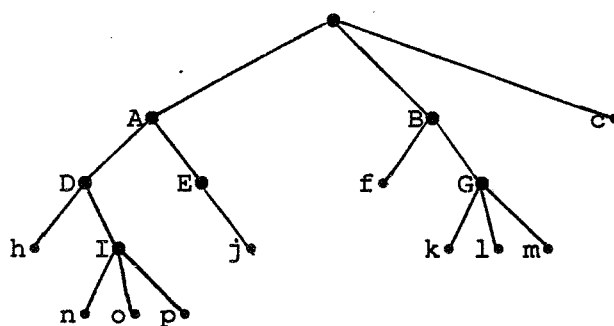
Le système TFMM définit un modèle virtuel de système de gestion de fichiers, appelé **système de fichiers uniformisé (Unified File System)**. L'UFS inclut en particulier, les notions de nature de fichier et les désignations universelles de fichiers présentées dans le chapitre précédent.

##### 2.1 DESCRIPTION

###### 2.1.1 Arborescence et désignation

L'organisation de l'ensemble des fichiers est une structure arborescente de répertoires et de fichiers.





A, B, D, E, G, I : répertoires  
 c, f, h, j, k, l, m, n, o, p : fichiers

Figure 3.1 : Exemple d'UFS

Un **nom UFS** est le nom associé à un fichier ou à un répertoire de l'arborescence UFS (exemples : A ; f ; h).

Une **désignation absolue** est la suite des noms UFS qui commence à la racine de l'arborescence et qui se termine au fichier ou au répertoire auquel elle est associée (exemples : A/D/h ; c).

Une **désignation relative** est la suite des noms UFS qui commence à un répertoire quelconque, appelé **base** et qui se termine au fichier ou au répertoire auquel elle est associée (exemple : D/h).

Les désignations de fichiers permettent de nommer de façon uniforme et non ambiguë les fichiers se trouvant sur des sites hétérogènes.

### 2.1.2 Attribut nature de fichier

L'attribut nature associe à un fichier la signification de son contenu. L'attribut nature est conservé pendant le transfert du fichier. Les natures de fichiers traitées actuellement sont la nature texte et la nature indéterminée.

### 2.1.3 Autres attributs

D'autres attributs sont associés au fichier UFS : des dates, un utilisateur propriétaire. Ils traduisent directement les attributs gérés par la plupart des SGF réels.

## 2.2 CORRESPONDANCE UFS-SGF

L'un des rôles de TFMM, est de mettre en correspondance l'UFS virtuel, avec les SGF réels.

Il s'agit de faire correspondre les désignations UFS avec des désignations SGF et les natures de fichiers UFS avec des fichiers réels (organisation, types de données, codage). Ces correspondances sont à la charge des réalisateurs de TFMM.

On peut remarquer que le modèle UFS correspond dans sa structuration (arborescence) à un grand nombre de SGF actuels et à la plupart des plus récents. La correspondance est donc triviale dans la plupart des cas : Unix, Multics, VAX-VMS, GCOS, MSDOS. Cependant le modèle peut également représenter des systèmes plus rudimentaires : système de gestion de fichiers à un ou deux niveaux de niveau. Il peut même convenir pour les fichiers partitionnés du type IBM. Il suffit de faire correspondre un répertoire UFS contenant uniquement des fichiers avec un fichier partitionné.

### 3 OPERATIONS ET RESULTATS D'OPERATION

Les services de transfert de fichiers sont mis à la disposition des utilisateurs sous la forme d'**opération** et de **résultat d'opération**. Les termes **initiateur** et **impliqué** associés aux utilisateurs sont utilisés par extension pour désigner les sites, les fichiers, etc.

#### 3.1 OPERATIONS

L'utilisateur dispose de trois opérations. Nous présentons ici leur rôle et leur fonctionnement. Pour plus de détails, on se reportera au chapitre 4 et à l'annexe A.

##### 3.1.1 Envoi de fichier

L'opération d'**envoi de fichier** permet à un utilisateur d'envoyer la copie d'un fichier local à un ou plusieurs utilisateurs distants.

Le fichier est copié, codé suivant sa nature. Il est envoyé dans un message, accompagné de paramètres tels qu'un commentaire et une référence, vers le(s) site(s) impliqué(s). En tant qu'utilisateur responsable du fichier de départ, l'utilisateur initiateur doit posséder des droits locaux suffisants pour lire le fichier.

Sur chaque site distant, un nouveau fichier est créé, et, c'est l'utilisateur impliqué qui en devient propriétaire. En tant qu'utilisateur responsable du fichier d'arrivée, l'utilisateur impliqué doit posséder des droits locaux suffisants pour créer le fichier. Il faut également qu'il ait préalablement donné son accord sous la forme d'une **autorisation d'arrivée**. Les autorisations d'arrivée sont présentées dans la section 4.

##### 3.1.2 Prise de fichier

L'opération de **prise de fichier** permet à un utilisateur d'aller chercher la copie d'un fichier se trouvant sur un site distant. Un premier message contenant la demande est envoyé vers le site impliqué. Un second message contenant le fichier en revient.

L'utilisateur impliqué doit avoir préalablement donné son accord sous la forme d'une **autorisation de départ**. Les autorisations de départ sont présentées dans la section 4. En tant que responsable du fichier de départ, l'utilisateur impliqué doit également posséder les droits locaux suffisants pour lire le fichier.

L'utilisateur initiateur devient le propriétaire du fichier. En tant qu'utilisateur responsable du fichier d'arrivée, il doit posséder des droits locaux suffisants pour créer le fichier.

### 3.1.3 Faisabilité d'envoi

L'opération **faisabilité d'envoi** permet à un utilisateur de vérifier si une opération d'envoi de fichier aboutirait dans les conditions actuelles. Il s'agit de vérifier l'adressage et l'autorisation d'arrivée. Cela évite d'envoyer inutilement des fichiers de taille importante. Comme l'opération d'envoi de fichier, l'opération de faisabilité d'envoi met en scène un utilisateur initiateur et un ou plusieurs utilisateurs impliqués.

Les deux opérations d'envoi et de faisabilité sont lancées à l'initiative d'utilisateur. Elles sont indépendantes. De plus, un résultat de faisabilité d'envoi positif ne garantit pas la réussite d'une opération d'envoi de fichier qui le suivrait. En effet, entre-temps, l'autorisation d'arrivée a pu être supprimée par l'utilisateur impliqué.

### 3.2 RESULTATS D'OPERATION

Comme leur nom l'indique, les **résultats d'opération** informent l'utilisateur initiateur du résultat d'une opération qu'il a préalablement démarrée. Etant données les caractéristiques du système de communication par transfert de messages, le résultat d'une opération intervient de façon asynchrone. En effet, il faut attendre un second message contenant le compte-rendu du site distant. Il arrivera dans l'heure et le jour qui suit, et cela en fonction des disponibilités du réseau et la priorité de transfert demandée par l'utilisateur.

Un résultat d'opération concerne un seul utilisateur impliqué. L'utilisateur initiateur recevra donc autant de résultats qu'il y a d'utilisateurs impliqués dans l'opération.

#### 3.2.1 Résultat d'envoi de fichier

Un **résultat d'envoi de fichier** rend compte d'une opération d'envoi de fichier. Il peut être positif ou négatif.

Dans le premier cas, il signifie que le fichier est bien parvenu à destination, que l'autorisation d'arrivée donnée par l'utilisateur impliqué correspond et que le fichier a été créé. Dans le second cas, cela vient soit de problèmes de transfert provenant des niveaux inférieurs (adressage incorrect, système de communication en panne...), soit de problèmes sur le site impliqué (autorisation d'arrivée refusée, droits locaux de l'utilisateur impliqué insuffisants pour créer le fichier).

Si un utilisateur ne veut pas recevoir les résultats associés à une opération d'envoi de fichier, il peut le préciser lors du démarrage de celle-ci.

#### 3.2.2 Résultat de prise de fichier

Un **résultat de prise de fichier** rend compte d'une opération de prise de fichier. Il peut être positif ou négatif.

Dans le premier cas, il signale que la demande est arrivée à destination, que l'autorisation de départ donnée par l'utilisateur impliqué correspond, que le second message contenant le fichier est arrivée à destination et que le fichier a été créé sur le site initiateur. Dans le second cas, cela vient soit de problèmes de transfert, soit de problèmes sur le site impliqué (autorisation de prise refusée, droits locaux de l'utilisateur impliqué insuffisants pour lire le fichier...), soit de problèmes sur le site initiateur (droits locaux de l'utilisateur initiateur insuffisants pour créer le fichier).

Comme pour l'envoi de fichier, l'utilisateur peut demander à ne pas recevoir de résultat. Ce qui n'empêche pas l'opération de prise de fichier d'être menée à son terme et au fichier d'être créé.

### 3.2.3 Résultat de faisabilité d'envoi

Contrairement aux opérations d'envoi et de prise de fichier, il y a toujours des résultats de faisabilité d'envoi. Ces derniers sont positifs ou négatifs.

Dans le premier cas, le message est bien parvenu à destination et l'autorisation d'arrivée donnée par l'utilisateur impliqué correspond. Dans le second cas, des problèmes de transfert sont apparus ou l'autorisation d'arrivée ne correspond pas.

## 4 AUTORISATIONS

### 4.1 ASSOCIATION OPERATION-AUTORISATION

Pour qu'une opération aboutisse, il faut que l'utilisateur impliqué ait donné préalablement son accord sous la forme d'une **autorisation**.

Une **autorisation d'arrivée** est associée à une opération d'envoi de fichier. Elle sera également consultée lors d'une opération de faisabilité d'envoi. Une **autorisation de départ** est associée à une opération de prise de fichier. On rappelle que les autorisations correspondent aux contrôles effectués sur les sites impliqués.

### 4.2 CHOIX

Plusieurs possibilités se présentent quant à la forme d'une autorisation. Doit-elle être valable une seule fois ou pendant une certaine durée ? Doit-elle concerner un ou plusieurs utilisateurs ? Doit-elle concerner un fichier ou un ensemble de fichiers ? Comment l'utilisateur initiateur d'une opération désigne-t-il l'autorisation associée ?

Le mécanisme doit être assez souple à utiliser. La forme d'une autorisation doit également être choisie en fonction des différentes utilisations envisagées de TFMM. Ainsi les groupes d'utilisateurs répartis sur des sites distants travaillant à la réalisation d'un projet commun, apparaissent comme des utilisateurs potentiels de notre application. Il faut donc essayer de faciliter leur travail, sans que cela s'oppose à une utilisation plus ponctuelle de TFMM.

### 4.3 DESCRIPTION

Une autorisation d'arrivée ou de départ est donnée par un utilisateur. Elle est valable jusqu'à sa suppression par le même utilisateur.

Une autorisation est désignée par un **nom**. Le nom est donné par l'utilisateur au moment de sa création. Au lancement d'une opération, l'initiateur doit fournir le nom de l'autorisation correspondante qui fait ainsi, fonction de mot de passe.

Une autorisation concerne un **ensemble d'utilisateurs** et un **ensemble de fichiers** se trouvant dans une **zone visible**.

Une zone visible est une sous-arborescence de l'UFS dans laquelle se trouvent les fichiers autorisés à être copiés dans le cas d'une autorisation de départ ou à être créés dans le cas d'une autorisation d'arrivée.

Dans une zone visible, l'utilisateur autorise un certain nombre de fichiers. Il peut les désigner en indiquant leur nom, ou celui d'un répertoire les contenant. Il peut également autoriser toute la zone visible. On peut ainsi donner libre accès à toute l'arborescence UFS-SGF, si la zone visible est égale à la racine et, si tous les

fichiers de la zone sont autorisés.

Voici quelques exemples d'autorisation.

```
nom d'autorisation: projet-Truc
zone visible: A
utilisateurs autorisés: /France/PTT/CNET/Dupond
                        /France/PTT/INRIA/*
                        /France/PTT/EMSE/Martin
fichiers autorisés: D/h
répertoires autorisés: D/I
```

Figure 3.2 : Exemple d'autorisation de départ  
(cf Figure 3.1 pour l'UFS)

```
nom d'autorisation: Cosac
zone visible: B
utilisateurs autorisés: /France/PTT/CNET
fichiers autorisés: tous
répertoires autorisés: tous
```

Figure 3.3 : Exemple d'autorisation d'arrivée  
(cf Figure 3.1 pour l'UFS)

L'utilisateur initiateur doit fournir, lors de la demande d'opération, la désignation relative du fichier impliqué par rapport à la zone visible (exemple : opération de prise de fichier D/I/n). De cette façon, il n'a pas à connaître la désignation absolue du fichier impliqué.

La notion de zone visible correspond bien à celle de groupe de travail. En effet, un utilisateur, participant à un groupe, regroupera les fichiers correspondants au même endroit. De plus, il arrive souvent dans les systèmes d'exploitation, que l'on ait à déplacer des sous-arborescences complètes. Il suffit alors à l'utilisateur de modifier la zone visible dans l'autorisation.

Le mécanisme d'autorisation en deux temps (zone visible, fichiers et/ou répertoires) concilie à la fois souplesse pour les utilisateurs et sécurité pour le système.



#### 4.4 CONTROLE

Pour qu'une opération réussisse, il faut, d'une part, une autorisation de l'utilisateur impliqué et, d'autre part, que ce dernier ait les droits locaux adéquats sur le fichier concerné.

Les droits locaux de l'utilisateur impliqué ne sont pas vérifiés au moment de la création d'autorisation, mais au moment de la copie ou de la création effective du fichier. En effet, les droits locaux de l'utilisateur impliqué peuvent avoir changé entre-temps.

## 5 SIGNALISATION

### 5.1 FONCTIONNALITES

On vient de voir que les utilisateurs impliqués donnent leur accord à l'exécution d'opérations, sous forme d'autorisations. Chaque autorisation concerne un ensemble d'utilisateurs et un ensemble de fichiers. Elle est valable jusqu'à sa suppression. Dans ces conditions, une autorisation peut donc donner lieu à une série d'opérations portant sur différents fichiers et concernant plusieurs utilisateurs distants.

Dans ces conditions, il paraît indispensable de prévenir un utilisateur lorsqu'une opération a lieu. Ceci est fait sous la forme d'une **signalisation**. De cette façon, rien ne se passe sans que l'utilisateur ne le sache, ce qui contribue à augmenter la sécurité. Les signalisations peuvent être également utilisées pour gérer une facturation ou faire des statistiques.

### 5.2 DESCRIPTION

On a deux types de signalisation. La **signalisation d'envoi de fichier** prévient un utilisateur qu'un fichier lui a été envoyé et qu'il en est devenu propriétaire. La **signalisation de prise de fichier** prévient un utilisateur qu'un fichier lui a été pris.

Les signalisations peuvent être négatives. Cela indique que les opérations d'envoi de fichier ou de prise de fichiers ont échoué. Parmi les paramètres fournis à l'utilisateur lors d'une signalisation, on trouve le nom de l'utilisateur initiateur, le nom de l'autorisation correspondante et la désignation du fichier pris ou créé.

Il n'y a pas de signalisation de faisabilité d'envoi. Cela ne paraissait pas très intéressant, dans la mesure où il n'y a aucune modification du SGF. De plus, une opération de faisabilité d'envoi est généralement suivie par une opération d'envoi de fichier, qui donnera lieu à une signalisation d'envoi.

## 6 ARCHITECTURE

### 6.1 MODELE FONCTIONNEL

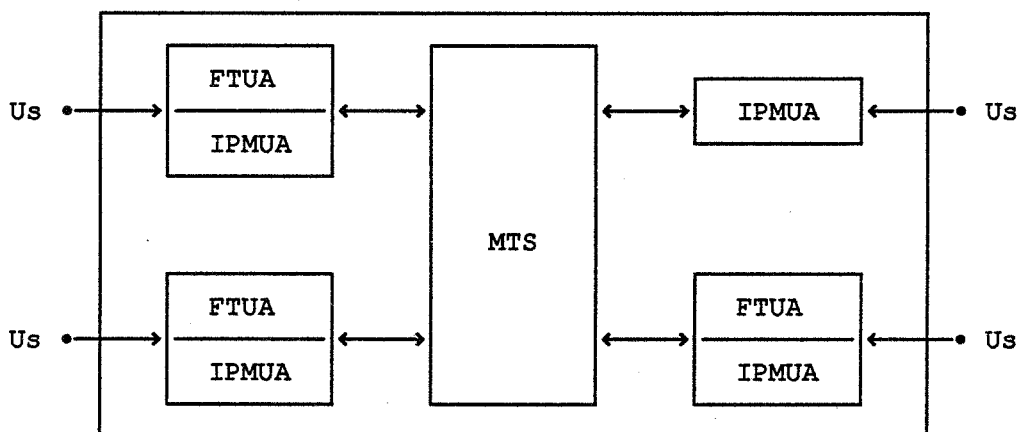
Les concepts présentés ci-dessus correspondent aux différents services rendus aux utilisateurs par le système TFMM. Le système TFMM utilise les services offerts par le Système de Transfert de Messages (MTS) défini par la série des recommandations X400 [3].

Un Agent Utilisateur (UA) est défini, dans la recommandation X400, comme l'entité fonctionnelle qui utilise les services de Transfert de Messages fournis par le MTS, afin de rendre des services à l'utilisateur auquel il est associé. Un UA est désigné par un nom d'O/R.

Les UA sont regroupés en classe d'après le type de contenu de messages (protocole) qu'ils peuvent traiter. Les UA d'une même classe sont appelés UA coopérants. Le Système de Messagerie de Personne à Personne (IPMS) fournit à un individu des services lui permettant de communiquer avec d'autres individus [6]. Les UA qui rendent les services d'IPM, constituent une classe particulière d'UA appelé UA d'IPM ou IPMUA.

On définit une nouvelle classe d'UA, rendant les services de Transfert de Fichiers en Mode Messagerie, appelé Agent Utilisateur de Transfert de Fichiers (FTUA). Les FTUA offrent aux utilisateurs l'ensemble des services dûs au titre de Transfert de Fichiers en Mode Messagerie, à savoir : les opérations, les résultats d'opération, les autorisations, les signalisations et la vision UFS du SGF.

On retrouve dans la figure 3.3 les différentes entités fonctionnelles présentées ci-dessus.



MTS : Système de Transfert de Messages

FTUA : Agent Utilisateur de Transfert de Fichiers

IPMUA : Agent Utilisateur de Messagerie de Personne à Personne

Us : Utilisateur des services offerts par le FTUA ou l'IPMUA

Figure 3.3 : Schéma fonctionnel du système Transfert de Fichiers en Mode Messagerie et de Messagerie de Personne à Personne

### Configuration physique

Contrairement au système d'IPM (cf Chapitre 2, section 2.3), on ne trouve pas de FTUA autonome dans le système TFMM. En effet, il faut que les fichiers et le FTUA soient sur le même site.

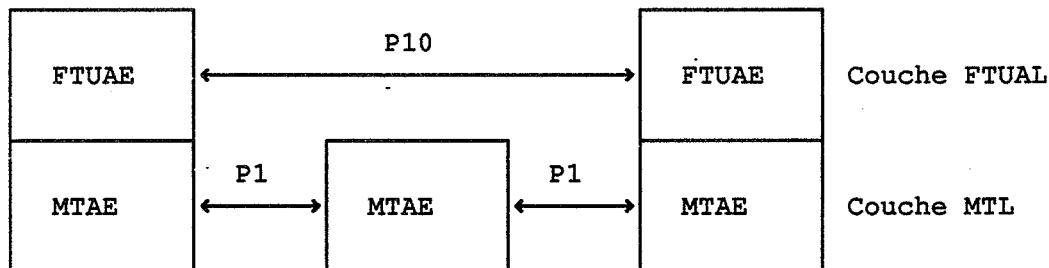
### 6.2 MODELE EN COUCHES

On adopte pour le système TFMM, le même découpage en couches que dans la messagerie X400. Pour la raison invoquée ci-dessus, il n'y pas d'entités de Dépôt et de Remise (SDE).

Les fonctions MTA se situent dans la couche MTL. Les entités MTAE sont les entités réalisant sur chaque site les fonctions MTA. Les entités MTAE communiquent à l'aide du protocole P1.

Les fonctions FTUA se trouvent dans la couche FTUAL. Les entités FTUAE sont les entités réalisant pour chaque utilisateur, les fonctions FTUA. Les entités FTUAE communiquent à l'aide du protocole P10.

Les couches MTL et FTUAL se trouvent dans la couche Application du modèle de référence OSI.



MTAE : Entité d'Agent de Transfert de Messages

FTUAE : Entité d'Agent Utilisateur de Transfert de Fichiers

Figure 3.4 : Modèle en couches

### 6.3 NOM D'O/R

Dans les différentes formes de nom d'O/R offertes actuellement par la messagerie X400, seule la forme 1, variante 1 convient pour désigner un FTUA. En effet, les autres formes servent à désigner les utilisateurs du service Telex et des autres services Télématicques. Un domaine de gestion auquel appartient un FTUA peut rendre obligatoire un certain nombre d'attributs (par exemple : le nom de domaine privé et le nom de personne sont obligatoires dans COSAC).

Forme 1, variante 1 :

- Nom de pays
- Nom de MD Administratif
- [Nom de MD privé]
- [Nom de personne]
- [Nom d'organisation]
- [Noms d'unité organisationnelle]
- [Attributs définis par un domaine]

Dans TFMM, on utilise les noms d'O/R pour désigner d'une part, les utilisateurs impliqués, d'autre part, les utilisateurs autorisés. L'association d'un nom d'O/R avec un utilisateur enregistré dans le système (login), est effectuée sous la forme d'un annuaire local.

## Chapitre 4

# COUCHE AGENT UTILISATEUR DE TRANSFERT DE FICHIERS

## 1 INTRODUCTION

Les services offerts par le système TFMM sont rendus par la coopération des entités d'Agent Utilisateur de Transfert de Fichiers (FTUAE) qui composent la couche FTUAL. Pour décrire la couche FTUAL, on décrit successivement :

- les services rendus par la couche FTUAL,
- l'interface supérieure offerte par les entités FTUAE à leur utilisateur,
- le protocole P10 utilisé par les entités FTUAE pour communiquer,
- les actions, c'est-à-dire la façon dont les FTUAE fonctionnent.

Cette description s'inspire directement de celles faites dans la série X400 pour décrire la couche Transfert de Messages [5] et la couche Agent Utilisateur de Transfert de Fichiers [6]. Le document décrivant de façon complète la couche FTUAL, se trouve dans l'annexe A.

### 1.1 METHODE DE DESCRIPTION DES SERVICES

Les services sont divisés en éléments de services. A chaque élément de service est associé un nom et un rôle. Pour décrire ce rôle, on ne dispose pas de formalisme particulier.

### 1.2 METHODE DE DESCRIPTION DE L'INTERFACE SUPERIEURE

L'interface supérieure est généralement décrite sous la forme de primitives de service. Une primitive de service correspond à une interaction entre l'utilisateur et l'entité. Elle demande ou fournit, un ou plusieurs éléments de service.

On a pour habitude de ne pas décrire l'interface supérieure de la couche la plus haute. Mais si la couche FTUAL se situe actuellement dans la couche la plus haute, rien n'interdit d'envisager de rajouter une nouvelle couche utilisant ses services. C'est pourquoi, nous avons choisi de décrire l'interface supérieure de la couche FTUAL.

On définit trois types de primitives.

- La primitive de **DEMANDE** permet à un utilisateur de demander à l'entité FTUAE, un service offert par le système TFMM.
- L'entité FTUAE répond à l'utilisateur ayant émis une primitive de **DEMANDE**, par une primitive de **CONFIRMATION DE DEMANDE**. Elle indique de cette façon qu'elle accepte ou refuse la **DEMANDE**. Les deux primitives **DEMANDE** et **CONFIRMATION DE DEMANDE** sont donc liées. On les représente sous la forme d'un schéma du type de celui de la figure 5.1.
- L'entité FTUAE dispose d'une primitive d'**INDICATION** pour fournir à l'utilisateur un service offert par TFMM. La primitive d'**INDICATION** est représentée sous la forme d'un schéma du type de celui de la figure 5.2.

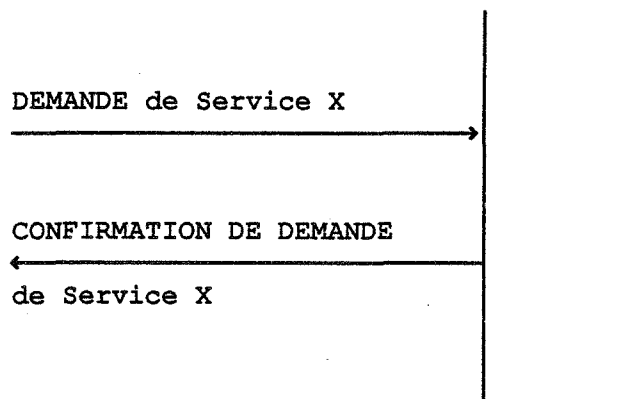


Figure 5.1 : Primitives de **DEMANDE** et de **CONFIRMATION de DEMANDE**

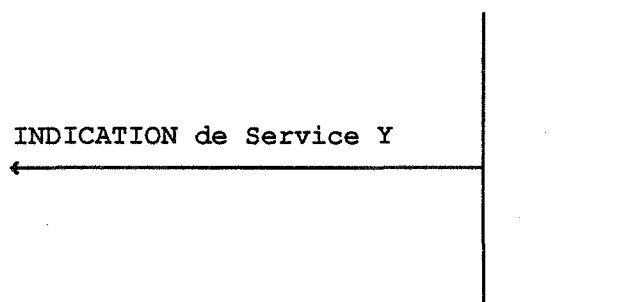


Figure 5.2 : Primitive d'**INDICATION**

On peut remarquer que ces trois types de primitives sont les mêmes que ceux utilisés dans la recommandation X411 pour décrire l'interface supérieure de la couche MTL.

Ces primitives sont adaptées à un mode de fonctionnement non connecté. Elles sont donc différentes des types de primitives définies dans le modèle de référence OSI, celles-ci étant adaptées à un fonctionnement connecté.

Les primitives de service permettent de décrire l'interface supérieure. Mais elles ne suffisent pas pour décrire le fonctionnement de la couche FTUAL. Les différentes relations qui existent, entre les primitives sont décrites dans les actions.

### 1.3 METHODE DE DESCRIPTION DU PROTOCOLE

Le protocole défini dans TFMM est décrit à l'aide de la syntaxe X409 définie dans la série X400 [4]. La syntaxe X409 fournit à la fois une méthode de description et une méthode de codage d'un protocole.

### 1.4 METHODE DE DESCRIPTION DES ACTIONS

Les actions décrivent le fonctionnement des entités FTUAE et, en particulier, les relations entre les services offerts (sous forme de primitives de service), le protocole et les services de la couche inférieure.

Les primitives MTL [5] offertes aux entités UAE, dont les FTUAE, sont les suivantes :

- La primitive de **Demande de Dépôt (DD)** permet à une entité UAE de demander l'envoi d'un message à une ou plusieurs entités UAE destinataires (cf Figure 5.3).
- L'entité MTAE répond à l'entité UAE ayant émis une primitive de **Demande de Dépôt**, par une primitive de **Confirmation de Demande de Dépôt (CDD)** (cf Figure 5.3).
- L'entité MTAE remet le message à une entité UAE destinataire à l'aide d'une primitive d'**Indication de Remise (IR)** (cf Figure 5.4).
- L'entité MTAE indique à l'entité UAE, si le message a été ou n'a pas été remis aux entités UAE destinataires, à l'aide d'une primitive d'**Indication d'Avis (IA)** (cf Figure 5.5).

Une opération TFMM utilise plusieurs messages de niveau MTL. Un premier message contient les paramètres de l'opération. Un ou plusieurs messages selon le nombre de FTUAE impliqués, contiennent les résultats de l'opération.



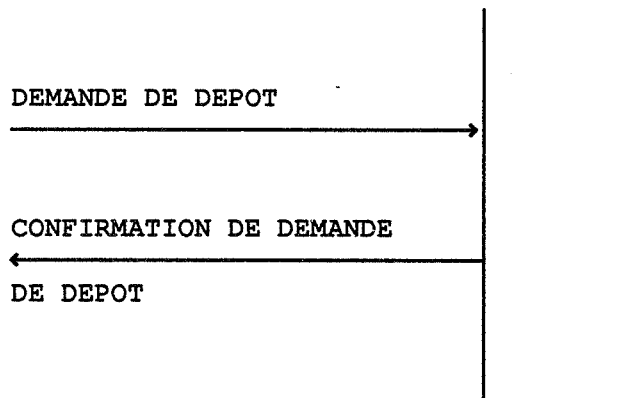


Figure 5.3 : Primitives de DEMANDE DE DEPOT et CONFIRMATION DE DEMANDE DE DEPOT

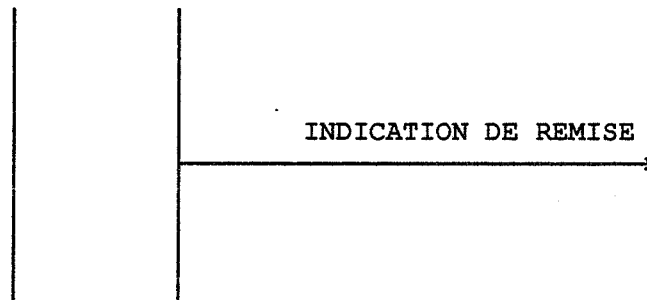


Figure 5.4 : Primitive d'INDICATION DE REMISE

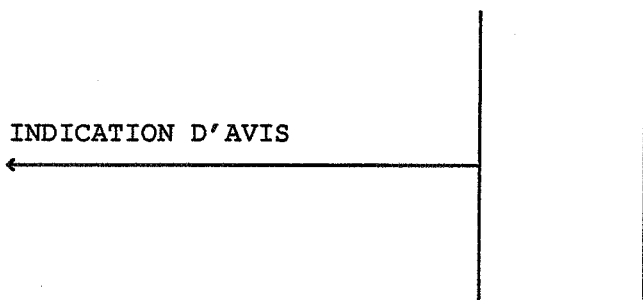


Figure 5.5 : Primitive d'INDICATION D'AVIS

On n'a pas de méthode particulière pour décrire les actions. Comme pour les services, on le fait de manière informelle. De plus, nous utilisons des schémas pour faciliter la compréhension.

## 2 SERVICES FTUAL

On retrouve les différents concepts décrits dans le chapitre 3 : les opérations d'envoi, de prise et de faisabilité, les résultats d'envoi, de prise et de faisabilité, la demande de résultat d'envoi de fichier, la demande de résultat de prise de fichier, les signalisations d'envoi et de prise, les dons et suppressions d'autorisations de départ et d'arrivée, les vérifications d'autorisations, les noms d'autorisations, la désignation de zone visible, les désignations des fichiers initiateur ou impliqué, la désignation de la nature de fichier, etc.

Nous décrivons, dans la section suivante, le rôle de certains éléments de service non introduits dans les concepts. L'ensemble des éléments de service est donné dans l'annexe A.

### 2.1 ROLE DE CERTAINS ELEMENTS DE SERVICE

#### 2.1.1 Commentaire

L'utilisateur initiateur en dispose pour commenter les opérations d'envoi de fichier et de prise de fichier. Un commentaire est un court message qui accompagne l'opération. Il est mis par l'entité FTUAE impliquée à la disposition de l'utilisateur impliqué lors de la signalisation.

exemple : Je t'envoie la version 2 du fichier codage.p.  
J'ai ajouté les fonctions ....

#### 2.1.2 Priorité de transfert

L'utilisateur peut choisir entre plusieurs priorités de transfert (urgent, normal, non-urgent) pour les opérations qu'il demande. C'est la priorité de transfert des messages MTS. Chaque priorité de transfert correspond à des délais et des coûts que l'utilisateur peut ainsi moduler en fonctions de ses besoins.

#### 2.1.3 Référence utilisateur

Un utilisateur initiateur utilise ce service pour référencer une opération qu'il a lancée. Cette référence est présente dans les signalisations et dans les résultats associés à l'opération.

#### 2.1.4 Identificateurs de demande et de signalisation

L'identificateur de demande est fourni par l'entité FTUAE initiatrice lors de la demande d'opération. On le retrouve dans les résultats associés. L'identificateur de signalisation est fourni par l'entité FTUAE impliquée lors de la signalisation.

Les identificateurs de demande sont tous différents les uns des autres. Il en va de même pour les identificateurs de signalisation. Ils permettent donc d'identifier de façon non ambiguë, toutes les interactions FTUAE-utilisateur, concernant les opérations. C'est indispensable en mode non connecté.

### **2.1.5 Dates associées au fichier copié**

Ce sont les dates associées au fichier copié : date de création, date de dernier accès et date de dernière modification. Dans la mesure où le SGF les gère, la couche FTUAL s'efforce de les communiquer.

### **2.1.6 Possibilités d'installation**

Le rôle d'une opération de faisabilité d'envoi est de vérifier l'adressage et l'autorisation d'arrivée. Dans le cas positif, l'entité FTUAE impliquée étudie, dans la mesure du possible, les autres conditions d'installation du fichier qui dépendent des droits de l'utilisateur impliqué et de la taille du fichier.

### **2.1.7 Renseignements d'installation**

Lorsqu'un fichier parvient après son transfert sur le site d'arrivée, son installation peut nécessiter la création d'un ou plusieurs répertoires. Ou bien, s'il existe déjà un fichier portant ce nom, le fichier est créé sous un autre nom généré automatiquement. On prévient alors les utilisateurs dans le résultat ou la signalisation.

### **2.1.8 Résultat anormal**

Par ce service, l'entité FTUAE initiatrice indique qu'il y a eu une anomalie consécutive à l'opération demandée. C'est une anomalie de niveau MTL, ou un problème de niveau FTUAL.

Les différentes anomalies de niveau MTL sont définies dans le document X411. Un problème de niveau FTUAL regroupe tous les cas d'anomalies, répertoriées ou non, pouvant survenir durant le fonctionnement d'une entité FTUAE.

### **2.1.9 Dates associées à une opération**

La date de demande est la date à laquelle l'utilisateur a demandée l'opération.

La date de résultat est la date à laquelle le résultat d'une opération est fournie, par l'entité FTUAE initiatrice, à son utilisateur. Il est intéressant de connaître cette date, sachant que l'utilisateur n'est pas forcément connecté au moment où le résultat arrive.

La date de traitement est la date à laquelle l'entité FTUAE impliquée a traité une opération. Cette date est fournie à l'utilisateur initiateur lors du résultat et à l'utilisateur impliqué lors de la signalisation.

La date de résultat anormal est la date à laquelle une erreur MTL ou un problème de niveau FTUAL ont été détectés.

Etant donné le fonctionnement en mode non connecté, toutes ces dates sont utiles pour suivre les différentes étapes d'une opération.

### 3 INTERFACE UTILISATEUR

Les éléments des services principaux tels que les opérations, les résultats, les signalisations, les dons et les suppressions d'autorisation sont associés à des primitives de service.

A chaque primitive est associée un certain nombre de paramètres. Les paramètres correspondent également à des éléments de service. La description complète de chaque primitive et de ses paramètres est faite dans l'annexe A.

Etudions les primitives de services liées à une opération d'envoi de fichier.

L'utilisateur émet une primitive de **Demande d'Envoi de Fichier (DEF)** (cf Figure 5.6), pour demander le service d'envoi de fichier. L'entité FTUAE répond à un utilisateur par une primitive de **Confirmation de Demande d'Envoi de Fichier (CDEF)** (cf Figure 5.6), qui indique si la demande est acceptée ou refusée.

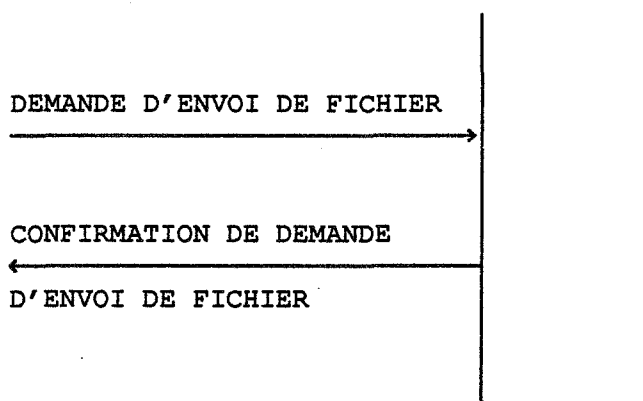


Figure 5.6 : Primitives de DEMANDE D'ENVOI DE FICHIER  
et de CONFIRMATION DE DEMANDE D'ENVOI DE FICHIER

Parmi les paramètres de la primitive DEF, l'utilisateur donne une référence (cf section 2.1.3). Dans ceux de la primitive CDEF, l'entité FTUAE rend l'identificateur de demande (cf section 2.1.4).

L'entité FTUAE émet une primitive d'Indication de Résultat d'Envoi de Fichier (IREF) pour rendre le service de résultat d'envoi de fichier (cf Figure 5.7). Parmi les paramètres de la primitive IREF, on trouve la référence utilisateur, l'identificateur de demande, la date du résultat (section 2.1.9) qui permettent à l'utilisateur reconnaître tous les résultats.

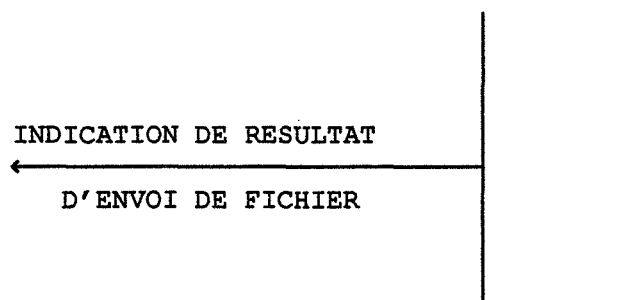


Figure 5.7 : Primitive d'INDICATION DE RESULTAT D'ENVOI DE FICHIER

L'entité FTUAE émet une primitive d'Indication de Signalisation d'Envoi de Fichier (ISEF) pour rendre le service de signalisation d'envoi de fichier (cf Figure 5.8). Parmi les paramètres de la primitive ISEF, on trouve également des identificateurs et des dates.

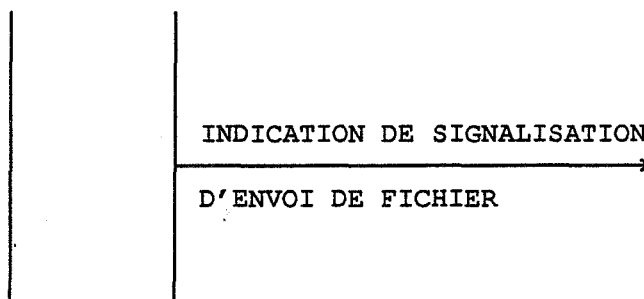


Figure 5.8 : Primitive d'INDICATION DE SIGNALISATION D'ENVOI DE FICHIER

## 4 PROTOCOLE P10

Une unité de données de protocole P10 est définie comme suit :

```
FTPDU ::= SEQUENCE {
    id-lancement [0] IMPLICIT Chaîne-IA5,
    [1] CHOICE {
        [0] IMPLICIT Lancement,
        [1] IMPLICIT Réponse,
        [2] IMPLICIT Problème }}

```

Les 3 catégories principales de types de données sont **Lancement**, **Réponse** et **Problème**.

Le composant **id-Lancement** est toujours présent. Il permet d'établir un lien entre les différents messages associés à une opération.

### 4.1 TYPE DE DONNEES LANCEMENT

Un type de données **Lancement** correspond au premier message envoyé associé à une opération.

```
Lancement ::= SEQUENCE {
    date-demande [0] IMPLICIT Date-Heure,
    [1] CHOICE {
        [0] IMPLICIT Lancement-Envoi,
        [1] IMPLICIT Lancement-Prise,
        [2] IMPLICIT Lancement-Faisabilité }}

```

Un type de données **Lancement** contient un certain nombre de composants correspondant à des éléments de service (référence utilisateur, désignations des fichiers, structure de fichier, date de demande...) et le fichier lui-même dans le cas d'un envoi de fichier.

Dans le cas de multideestination (envoi ou faisabilité), l'entité FTUAE initiatrice envoie à toutes entités FTUAE impliquées, le même message.

### 4.2 TYPE DE DONNEES REPONSE

Un type de données **Réponse** correspond au message renvoyé par l'entité FTUAE impliquée après réception et traitement d'un type de données **Lancement**.

```
Réponse ::= SEQUENCE {
    date-traitement [0] IMPLICIT Date-Heure,
    [1] CHOICE {
        [0] Réponse-Envoi,
        [1] Réponse-Prise,
        [2] Réponse-Faisabilité }}

```

Il n'y a pas de réponse dans le cas d'un envoi sans demande de résultat d'envoi et dans le cas d'une prise sans demande de résultat de prise ayant échoué.

Dans un type de données **Réponse-Envoi**, on trouve les composants correspondant au service de résultat d'envoi de fichier.

Dans un type de données **Réponse-Prise-Négative**, on trouve les composants correspondant au service de résultat de prise. Dans un type de données **Réponse-Prise-Positive**, on trouve également le fichier copié et les paramètres (nom, structure) permettant son installation. On a choisi de faire circuler ces paramètres dans le protocole ce qui évite à l'entité FTUAE de les stocker.

Dans une **Réponse-Faisabilité**, on trouve les composants correspondant au service de résultat de faisabilité.

Dans le cas de multidestination, il y aura donc autant de messages contenant des réponses qu'il y a d'utilisateurs impliqués.

### 4.3 TYPE DE DONNEES PROBLEME

Le type de données **Problème** est utilisé par l'entité FTUAE lorsqu'il détecte une anomalie au cours du traitement d'un type de données **Lancement**. Il permet de rendre les services de résultat.

```
Problème ::= SEQUENCE {
    date-problème [0] IMPLICIT Date-Heure,
    type-problème [1] IMPLICIT INTEGER {
        protocole-incorrupt (0),
        ressources-insuffisantes (1),
        autres (2) }}
```

### 4.4 ROLE DE CERTAINS TYPES DE DONNEES

#### 4.4.1 Type de Données Fichier

Ce type de données correspond au fichier codé suivant sa nature.

```
Fichier ::= CHOICE {
    [0] IMPLICIT Texte-Ligne,
    [1] IMPLICIT Flot}
```

Texte-Ligne ::= SEQUENCE of Ligne

Ligne ::= Chaîne-IA5

Flot ::= Octet-String



#### 4.4.2 Type de Données Chaîne-TFMM

On définit un type de données Chaîne-TFMM pour disposer d'une chaîne qui soit composée à partir d'un sous-ensemble de caractères IA5 dont on exclut les caractères non imprimables (caractères de contrôle). Ce type de données est utilisé dans les types de données Nom-UFS, Nom-Autorisation, Référence-Utilisateur, et Commentaire.

Chaîne-TFMM ::= [ APPLICATION 1] IMPLICIT Chaîne-IA5

#### 4.4.3 Types de Données UFS

Le type de données Désignation-UFS-Absolue est utilisé pour la désignation du fichier initiateur.

Désignation-UFS-Absolue ::= SEQUENCE of Nom-UFS

Le type de données Désignation-UFS-Relative est utilisé pour la désignation du fichier impliqué.

Désignation-UFS-Relative ::= SEQUENCE of Nom-UFS

Le type de données Nom-UFS est utilisé dans les types de données Désignation-UFS-Absolue et Désignation-UFS-Relative.

Nom-UFS ::= Chaîne-TFMM

## 5 ACTIONS FTUA

Le fonctionnement d'une entité FTUAE est décrit sous la forme d'actions. Une action ou une série d'actions est déclenchée par la réception d'une primitive de l'interface supérieure (primitives Utilisateur) ou de l'interface inférieure (primitives MTL).

L'ensemble des actions est donné dans l'annexe A. Etudions le cas d'une opération d'envoi de fichier avec demande de résultat et plusieurs utilisateurs impliqués.

On trouve à la fin de cette section un exemple de fonctionnement sous la forme d'un scénario.

### 5.1 ACTIONS LIEES A UN ENVOI DE FICHIER AVEC DEMANDE DE RESULTAT

On utilise des figures pour faciliter la compréhension. Dans toutes les figures, l'entité FTUAE initiatrice se trouve à gauche, et, les entités FTUAE impliquées se trouvent à droite. La première figure donne tous les enchaînements possibles des actions.

5.10 → 5.11 ou 5.12

5.12 → 5.13 ou 5.14 ou 5.15

5.15 → 5.16 ou 5.17

5.16 → 5.18 ou 5.19

5.17 → 5.18 ou 5.19

Figure 5.9 : Enchaînements des actions

L'utilisateur demande à l'entité FTUAE, un envoi de fichier à l'aide de la primitive Demande d'Envoi de Fichier (DEF) (cf Figure 5.10). Il fournit les différents paramètres de la primitive, parmi lesquels les noms d'O/R des utilisateurs impliqués, les noms des fichiers initiateur et impliqués, le commentaire, etc. Il indique qu'il veut recevoir les résultats associés.

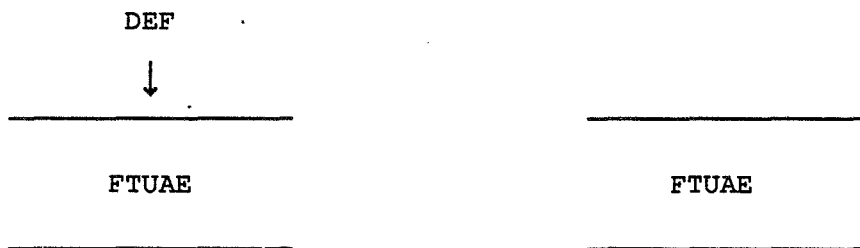


Figure 5.10

L'entité FTUAE vérifie les différents paramètres. Elle répond par une primitive de Confirmation de Demande d'Envoi de Fichier (CDEF) qui est positive (cf Figure 5.11) ou négative (cf Figure 5.12).

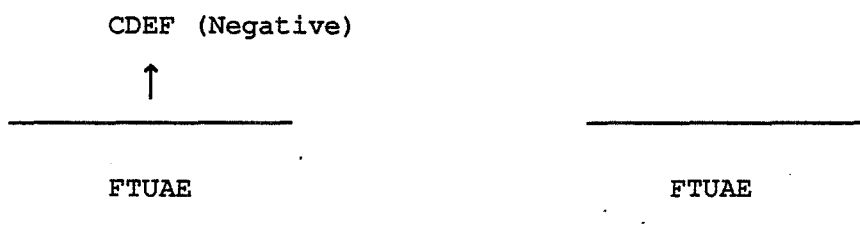


Figure 5.11

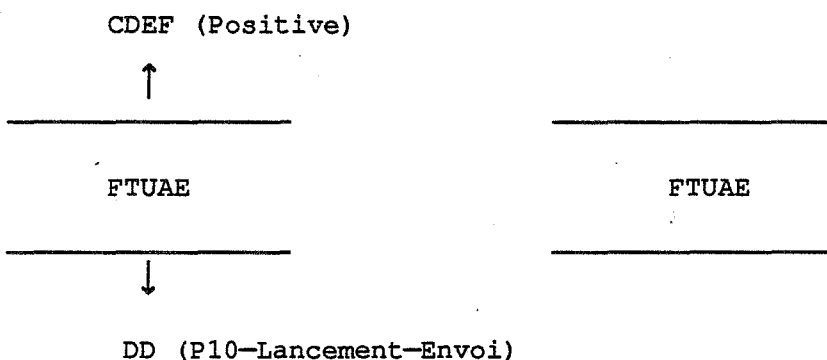


Figure 5.12

Dans le cas positif, l'entité FTUAE génère un élément de protocole P10 (Lancement-Envoi). Il émet auprès de MTS une primitive de Demande de Dépôt (DD) (cf Figure 5.12) dont les paramètres sont les suivants : les destinataires sont les entités FTUAE impliquées, le type de contenu est P10, le contenu est l'élément de protocole P10 généré, la priorité est la priorité de transfert indiquée par l'utilisateur.

L'entité MTAE répond à l'entité FTUAE par une primitive de Confirmation de Demande de Dépôt (CDD). Si la primitive de CDD est négative, l'entité FTUAE émet auprès de l'utilisateur des primitives d'Indication de Résultat d'Envoi de Fichier (IREF) de type anormal (cf Figure 5.13). Il y a autant de primitives IREF qu'il y a d'entités FTUAE impliquées.

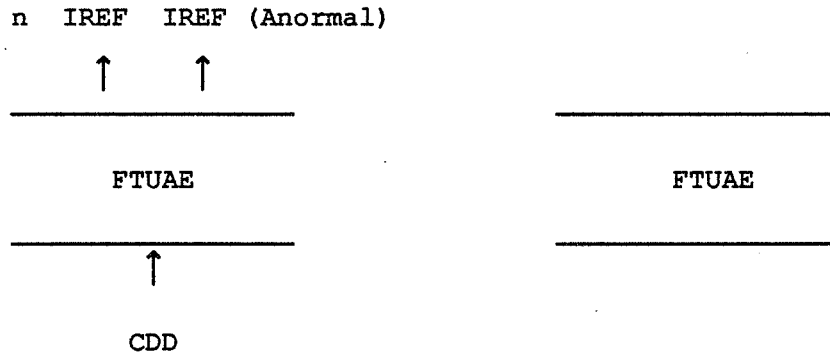


Figure 5.13

Si l'entité MTAE émet auprès de l'entité FTUAE, une primitive d'Indication d'Avis (IA), c'est pour lui indiquer que le message n'a pas pu être remis à une ou plusieurs entités FTUAE destinataires. L'entité FTUAE émet auprès de l'utilisateur, des primitives d'Indication de Résultat d'Envoi de Fichier (IREF) de type anormal (cf Figure 5.14). Il y a autant de primitives qu'il y a d'entités FTUAE concernées par la primitive d'IA.

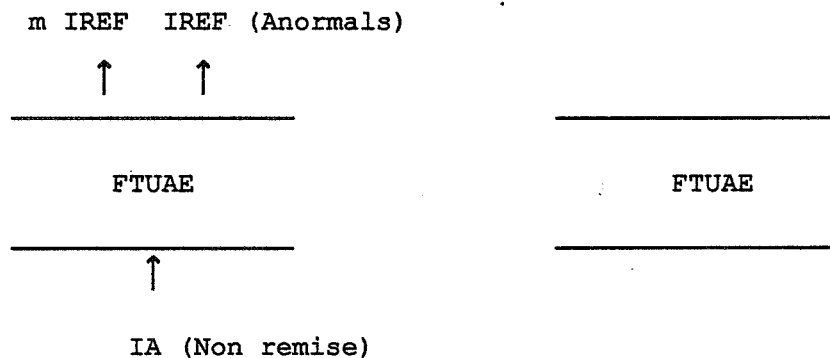


Figure 5.14

Dans le cas où tout se passe bien, chaque entité FTUAE impliquée reçoit une primitive d'Indication de Remise (IR) contenant un élément de protocole P10 (Lancement-Envoi) (cf Figure 5.15). Elle effectue le traitement associé. Elle vérifie l'autorisation d'arrivée correspondante. Elle vérifie les droits locaux de l'utilisateur impliqué. Elle crée le fichier.

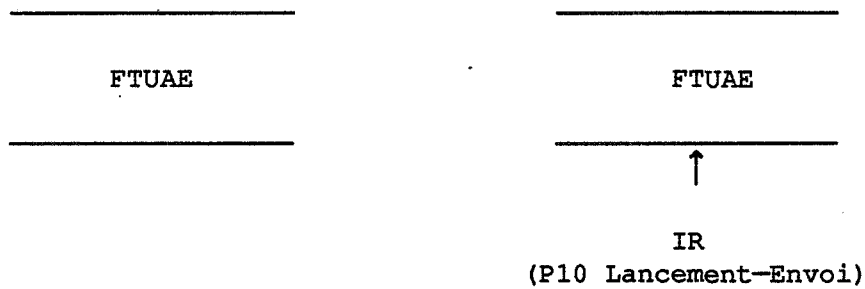


Figure 5.15

Suivant le résultat de ces traitements, l'entité FTUAE émet auprès de l'utilisateur, une primitive d'Indication de Signalisation d'Envoi de Fichier (ISEF) positive ou négative (cf Figure 5.16). Puis l'entité FTUAE génère un élément de protocole P10 Réponse-Envoi-Positive ou Réponse-Envoi-Négative (cf Figure 5.16).

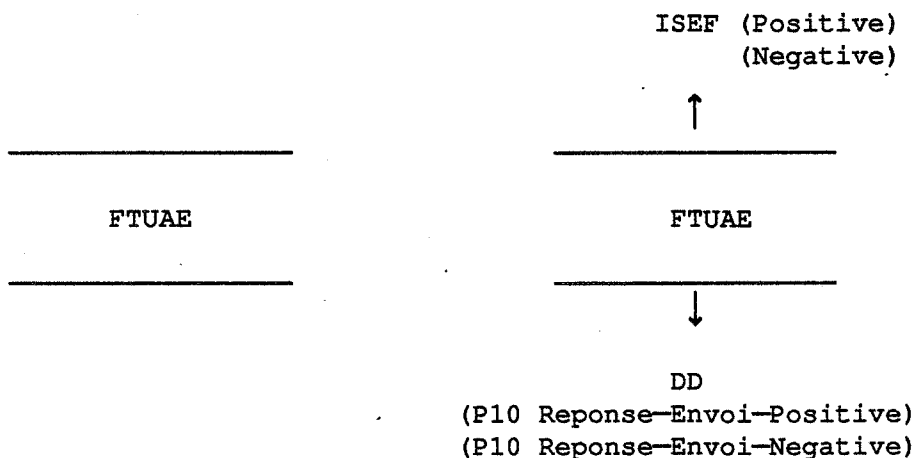


Figure 5.16

S'il se passe une anomalie au cours de ces traitements (par exemple : manque de ressources), elle génère un élément de protocole P10 Problème (cf Figure 5.17).

L'entité FTUAE impliquée émet ensuite, auprès de MTL, une primitive de Demande de Dépôt (DD) (cf Figures 5.16 et 5.17) dont les paramètres sont les suivants : le destinataire est l'entité FTUAE initiatrice, le type de contenu est P10, le contenu est l'élément de protocole P10 généré, l'avis de non remise est supprimé et la priorité est la priorité de transfert du précédent message.

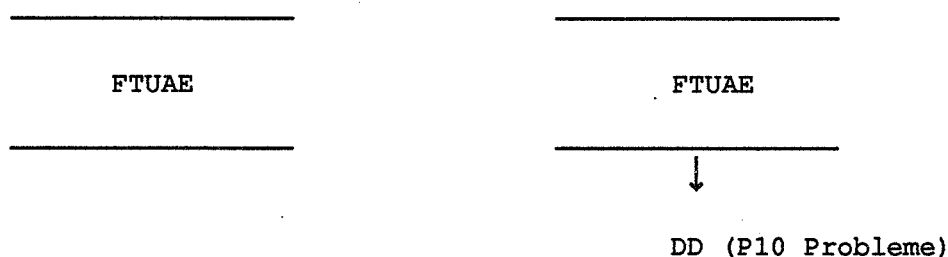


Figure 5.17

L'entité MTAE répond à l'entité FTUAE impliquée par une primitive de Confirmation de Demande de Dépôt (CDD) (cf Figure 5.18). L'entité FTUAE ne fait rien, l'utilisateur initiateur ne recevra jamais le résultat correspondant. On peut, dans une réalisation, décider de prévenir l'administrateur du logiciel.

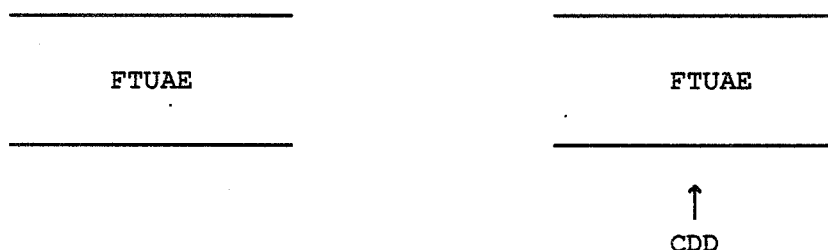


Figure 5.18

L'entité FTUAE initiatrice reçoit une primitive d'Indication de Remise (IR) contenant un élément de protocole P10. Elle émet auprès de l'utilisateur, une primitive d'Indication de Résultat d'Envoi de Fichier (IREF) de type positif, négatif ou anormal suivant la valeur de protocole P10 (cf Figure 5.19).

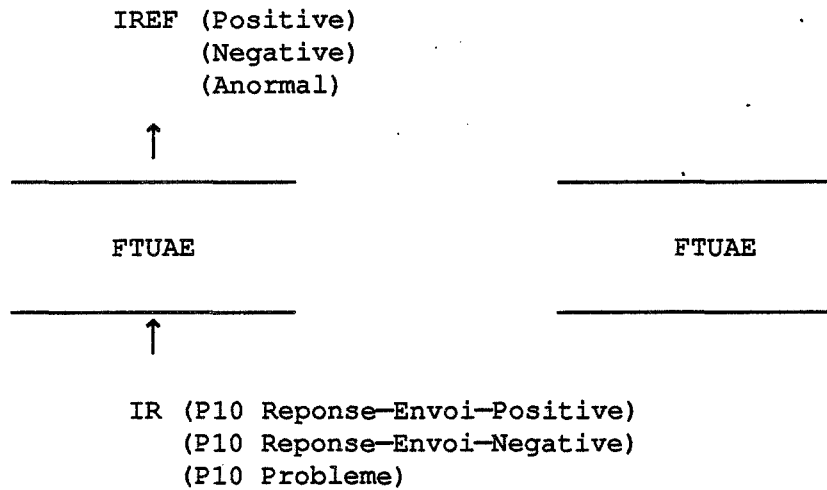


Figure 5.19

## 5.2 EXEMPLE

Voici sous la forme de scénario, une opération d'envoi de fichier avec demande de résultat.

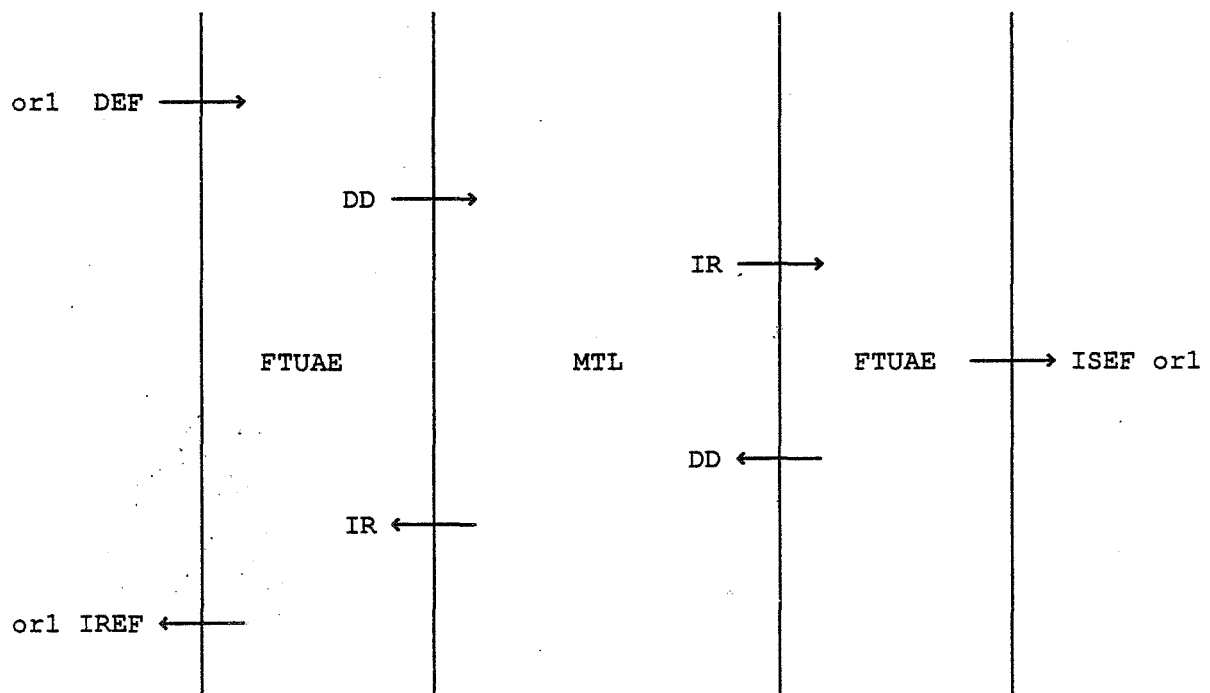


Figure 5.20

Emission d'une primitive DEF : l'utilisateur or1 demande l'envoi du fichier F1 à l'utilisateur or2. Le nom de l'autorisation est A2, et le nom du fichier à créer sur le site distant est f2. L'utilisateur or1 donne la référence R1.

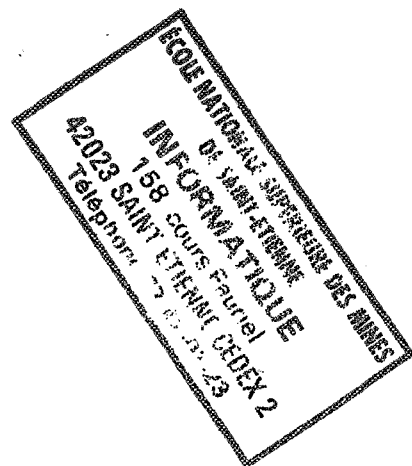
Action du FTUAE à la réception de la primitive DEF : prise de copie du fichier F1, fabrication d'un identificateur de lancement IL1, préparation du protocole P10, dépôt du message auprès de MTL grâce à une primitive DD, émission d'une primitive CDEF vers l'utilisateur or1 CDEF : (référence R1, identificateur de demande ID1, etc).

Le MTL achemine le message vers le site de l'utilisateur or2. Le message est remis au FTUAE de l'utilisateur or2 dans une primitive IR.

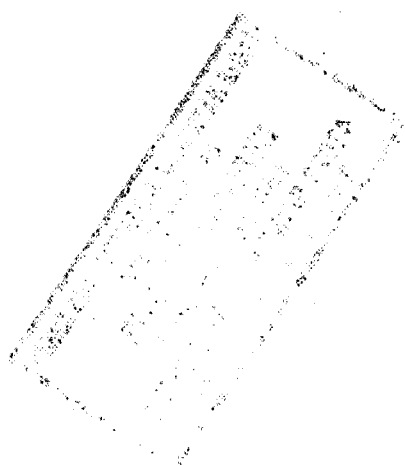
Actions du FTUAE à la réception d'une primitive IR : vérification de l'autorisation d'arrivée A2, déduction du nom absolu du fichier :  $F2 = \text{zone visible} + f2$ , vérification des droits de l'utilisateur or2 pour créer un fichier F2, création du fichier contenant le fichier transféré, émission d'une primitive ISEF vers l'utilisateur or2. ISEF : (date, or1, nom F1, référence R1, commentaire, nom d'autorisation A2, nom F2, etc), préparation du P10 de Réponse-Envoi-Positive (même identificateur de lancement IL1), soumission d'une primitive DD vers MTL.

Le MTL achemine le message vers le site de l'utilisateur or1. Le message est remis au FTUAE de l'utilisateur or1 dans une primitive IR.

Actions du FTUAE à la réception d'une primitive IR : émission vers l'utilisateur d'une primitive IREF. IREF : (date IREF, or2, référence R1, identificateur de demande ID1)







## **Chapitre 5**

### **PRESENTATION DE LA REALISATION**

#### **1 CADRE COSAC**

Le logiciel GTF est une réalisation du système TFMM. Il fait partie de l'ensemble COSAC-V5 développé par l'équipe COSAC du CNET dirigée par C.Kintzig. Le produit COSAC-V5 comprend, outre GTF, une réalisation du système d'IPM appelée GCE, une réalisation du système MTS appelée GM-TM, ainsi que des logiciels de taxation, d'annuaire et des adaptateurs à différentes messageries existantes.

L'ensemble COSAC-V5 est écrit en **Pascal standard** pour sa majeure partie. Il a été développé sous le système **Unix** et sur la machine **sm90**.

La licence COSAC-V5 a été vendue par le CNET à plusieurs sociétés dont Primatel et CIMSA.

GTF a été développé en parallèle sur Unix, par notre équipe de recherche, et sur Multics, par Patrice Fauvel qui travaille au CNET.

#### **2 SPECIFICATIONS DE REALISATION**

GTF est constitué d'un ensemble de modules et d'objets. Chaque module réalise une fonction particulière. Les modules correspondent à la partie dynamique de la réalisation. Ils sont activés soit par un utilisateur, soit par l'administrateur GTF, soit de façon automatique par le système. Les objets correspondent à la partie statique de la réalisation. Ils sont utilisés par les modules et ont une durée de vie supérieure à au temps d'activation des modules.

##### **2.1 OBJETS PRINCIPAUX**

On utilise les objets pour stocker des informations que l'on souhaite conserver, pour les fournir à l'utilisateur lorsqu'il le souhaite. C'est le cas des paramètres associées aux opérations, aux résultats et aux signalisations.

On conserve dans l'objet **DEMRES** les informations associés à une opération : les paramètres de la demande (date de demande, référence utilisateur, nature de fichiers, identificateur de demande) et ceux des résultats parvenus (date de résultat, raison du résultat négatif). Un utilisateur consulte un objet **DEMRES** pour

connaître l'état de l'opération qu'il a lancée. Un objet **BADR** (Boîte à Demande et Résultats) contient tous les objets **DEMRES** associés à un utilisateur.

Un objet **SIG** contient les paramètres d'une signalisation. Un objet **BAS** (Boîte à Signalisation) contient tous les objets **SIG** associés à un utilisateur. Un utilisateur consulte un objet **BAS** pour savoir quel fichier lui a été envoyé ou pris.

On stocke également sous forme d'objets les informations concernant les autorisations. Les objets **LAA** et **LAD** contiennent les autorisations données par un utilisateur. Ils sont consultés lorsqu'une demande d'opération arrive. Les utilisateurs peuvent les lire et les modifier.

L'objet **TRANS** contient des informations sur une opération lancée et pas encore terminée. Il permet de faire le lien entre la demande et les résultats. Lorsque tous les résultats sont parvenus, on détruit l'objet **TRANS**.

## 2.2 INTERFACE MTA

Comme **GCE** (réalisation **COSAC** d'un **IPMUA**), **GTF** s'interface avec le module **GM** qui est la réalisation de la partie supérieure de **MTAE** (sauf **RTS**).

L'interface **GM** est réalisée sous la forme de files d'attente. Une file d'attente **FAE** (File d'Attente Entrante) contient les messages en attente d'être traités par **GM**, donc déposés par **GTF** ou **GCE**. Deux files d'attente **FAS** (File d'Attente Sortante) contiennent les messages en attente d'être traités par **GTF** et **GCE**.

## 2.3 LES MODULES PRINCIPAUX

On retrouve sous forme de modules, les différents traitements effectués par l'entité **FTUAE**.

### 2.3.1 Le module DO

Le module **DO** (Demande d'Opération) correspond au traitement des primitives de Demande d'Envoi de Fichier, de Demande de Prise de Fichier et de Demande de Faisabilité d'Envoi. Il est activé par l'utilisateur.

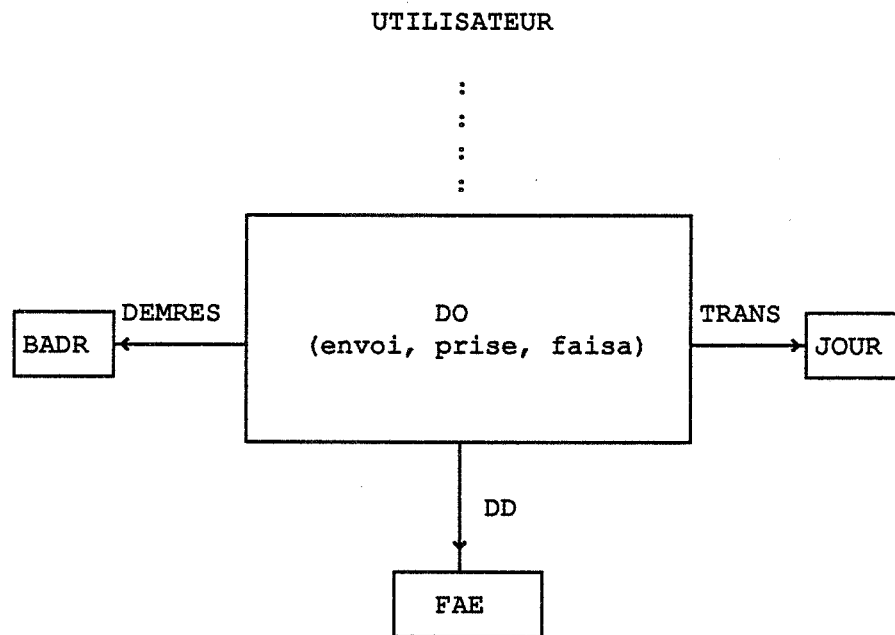


Figure 5.1 : Fonctionnement de DO

Une première partie du module DO constitue l'interface utilisateur. Elle réalise la saisie et la vérification des paramètres. La seconde partie correspond au traitement FTUAE à proprement parler. Elle contient la copie et le codage du fichier, la vérification des droits locaux de l'utilisateur, la génération du P10, la création des objets DEMRES et TRANS, et enfin, le dépôt d'un message à l'aide d'une primitive DD auprès de GM.

### 2.3.2 Le module GTFGM

Le module GTFGM correspond au traitement effectué par l'entité FTUAE à la suite de la réception de primitives MTL. Ces dernières se trouvent dans la file d'attente FAS de GTF. Il peut être activé soit par GM (MTAE) à chaque primitive, soit périodiquement par le système.

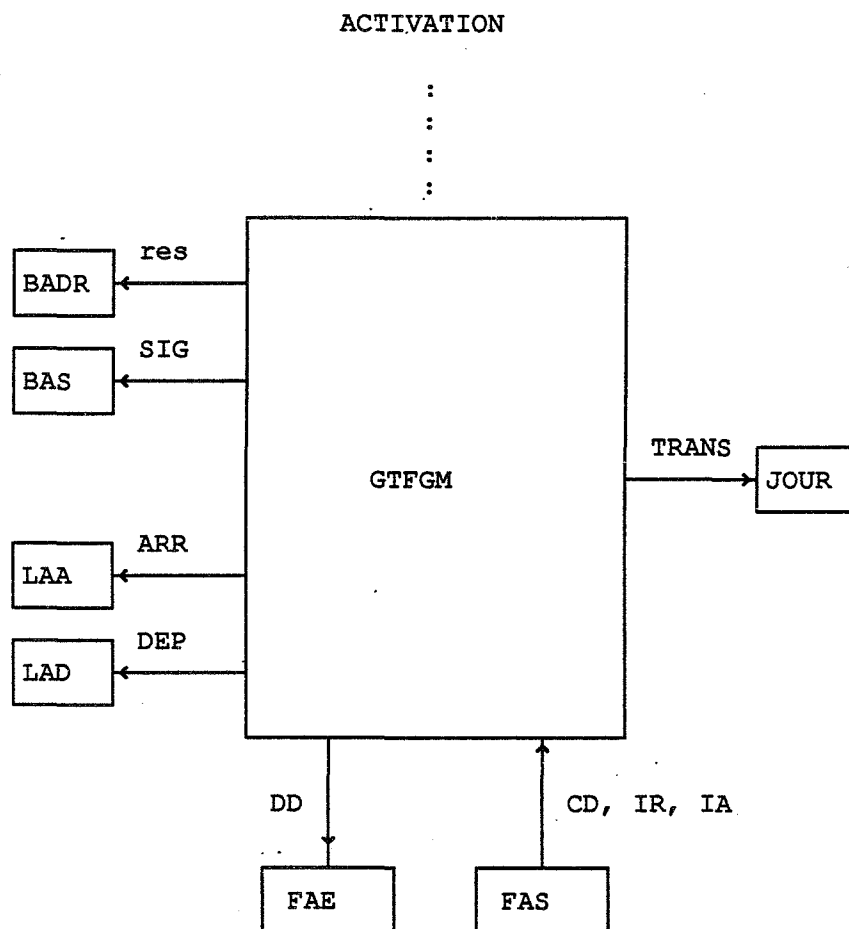


Figure 5.2 : Fonctionnement de GTFGM

Suivant les types de primitives MTL (CDD, IA, IR) et les catégories de P10 (Lancement, Réponse, Problème), GTFGM rajoute un ou plusieurs résultats dans l'objet DEMRES, consulte les objets LAA ou LAD, modifie l'objet TRANS, génère du P10 (Réponse ou Problème) et dépose une primitive DD, dans la file d'attente FAE.

### 2.3.3 Le module CB

Le module CB (Consultation des Boîtes) est utilisé par l'utilisateur pour consulter les boîtes BADR ou BAS. Il est constitué de deux sous-modules dont le fonctionnement est similaire.

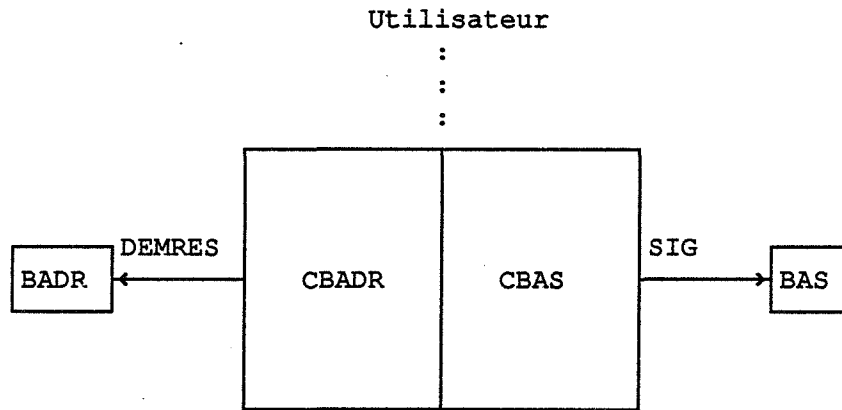


Figure 5.3 : Fonctionnement de CB

Le contenu des boîtes est présenté sous la forme d'un sommaire qui donne la référence utilisateur, les identificateurs de demande ou de signalisations et un numéro d'objet dans la boîte.

L'utilisateur peut à l'aide de ce numéro, demander à lire ou à détruire un objet DEMRES ou SIG. Il peut détruire un objet DEMRES dont tous les résultats ne sont pas parvenus. Lorsque les résultats arriveront, GTFGM les jettera.

Les destructions d'objets DEMRES et SIG sont effectives seulement la sortie du module. Les boîtes BAS et BADR, sont consultés par les modules DO, CB et GTFGM en exclusion mutuelle.

#### 2.3.4 Le module MJA

Le module MJA (Mise à Jour des Autorisations) correspond au traitement des primitives de don ou de suppression d'autorisations. Il est constitué de deux sous-modules dont le fonctionnement est identique. Il est activé par l'utilisateur.

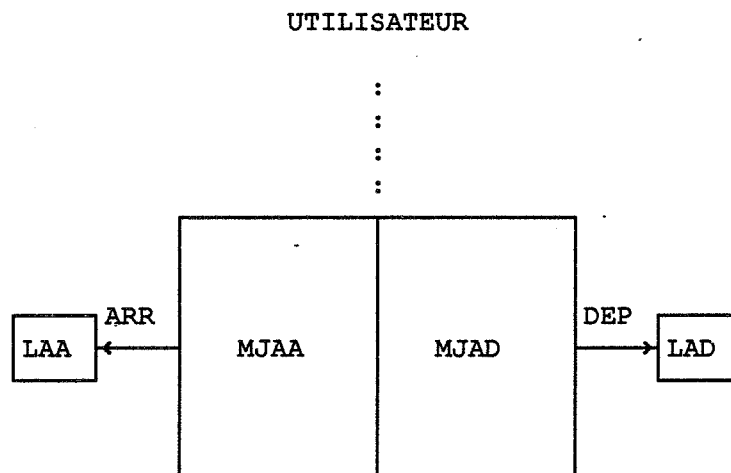


Figure 5.4 : Fonctionnement de MJA

Chacun des modules MJAA et MJAD présente à l'utilisateur les objets présents dans les listes sous la forme d'un sommaire qui donne le nom et le numéro de l'autorisation.

L'utilisateur peut lire, détruire, ajouter ou modifier des autorisations. Comme pour CB, les modifications sur les listes deviennent effectives à la sortie de MJA. L'accès aux listes entre MJA et GTFGM est géré par un mécanisme d'exclusion mutuelle.

### 2.3.5 Module de gestion

Le module GESGTF réalise la gestion de GTF. Il est activable par l'administrateur de GTF. Il permet de rajouter un nouvel utilisateur de GTF (création des objets associés). Il sert également à consulter les messages d'anomalie générés par les autres modules au cours de leur fonctionnement.

## 2.4 FONCTIONNALITES DEPENDANTES DU SYSTEME

Les fonctionnalités FTUAE qui dépendent du système, concernent principalement les correspondances UFS-SGF (désignation et nature), et les droits locaux des utilisateurs.

### 2.4.1 Désignations SGF-UFS

Sous Unix et Multics, cela ne pose aucun problème puisque dans les deux systèmes, les SGF sont organisés en arborescence. On interdit l'utilisation de noms de fichiers qui remontent l'arborescence.

Dans les modules CB, DT et MJA, on utilise les noms de fichiers SGF pour les fichiers locaux, et les désignations UFS pour les fichiers distants.

### 2.4.2 Nature de fichier

On traite, pour le moment, dans GTF uniquement les natures texte et indéterminée.

Sous Unix, la correspondance ne pose pas de problèmes. Pour les fichiers texte, une ligne est constituée de tous les caractères jusqu'au "Line Feed". Pour les fichiers de nature indéterminée, il suffit de lire tous les octets jusqu'à la fin de fichier.

### 2.4.3 Les droits locaux

Pour vérifier les droits d'un utilisateur, GTF prend momentanément l'identification de ce dernier. Il effectue les fonctions de lecture, écriture et création sur le fichier demandé. De cette façon, l'utilisateur devient le propriétaire du fichier créé.

## 3 REALISATION

Sous Unix, nous avons utilisé les possibilités offertes par le Pascal Modulaire et la gestion des interfaces de M. Prusker (GIPSI), et par l'utilitaire d'Unix Makefile. L'utilisation des fonctions système est réduite au minimum. Ce sont principalement des fonctions de gestion de terminal, des fonctions de gestion de fichiers (nom de fichier, dates de fichier, accès au contenu). Toutes les fonctions système sont chapeautées par une fonction Pascal, ce qui facilite le transport de GTF. L'ensemble de GTF comprend environ 250 fonctions dans 70 modules.





## CONCLUSION

Parmi les différents aspects abordés dans le système TFMM, il apparaît que les points les plus significatifs et les plus originaux sont d'une part la notion de nature de fichiers, et, d'autre part l'utilisation du mode messagerie X400. Je dirais que c'est ce qu'il faut retenir de TFMM.

Je ne reviendrais pas ici sur les avantages du mode messagerie X400. Bien qu'elle surprenne encore, c'est aujourd'hui une idée qui fait son chemin. La définition d'autres applications COSAC telles que les conférences réparties, l'exécution de jobs à distance et l'interrogation des bases de données ont démontré une nouvelle fois son utilité. L'erreur serait d'en faire l'unique méthode pour communiquer. C'est un nouveau moyen qu'il faut considérer au même titre que les autres lorsque l'on veut définir une application répartie.

La notion de nature de fichiers dépasse le cadre des communications, tous les fichiers ont une nature, même si elle n'est exprimée nulle part. Les fichiers de texte en sont l'exemple parfait. Ils ont une nature implicite que nous avons décrite à posteriori. Mais pour que le système TFMM soit facilement et largement utilisé, il faut que la nature du fichier et la syntaxe associée soient définies dès le départ. C'est la méthode qui a été employée pour la définition des documents ODA/ODIF. Les informations manipulées, mémorisées et échangées ont été définies de manière logique, indépendamment de toute réalisation. Une forme particulière sous laquelle les informations sont échangées a été décrite en même temps. Cette démarche apparaît indispensable aujourd'hui. En effet, il ne suffit pas de posséder des systèmes de communication hétérogènes pour communiquer, il faut également définir les informations échangées de façon logique et indépendante des réalisations.

Pour favoriser l'utilisation de TFMM, plusieurs évolutions sont prévues. Dès que de nouvelles natures de fichiers seront définies, on les rajoutera à TFMM. On veut également pouvoir traiter automatiquement n'importe quelle nature de fichier, pas forcément universellement connue mais définie dans un cadre plus limité. Pour cela, on offre la possibilité à l'utilisateur d'associer aux événements "copie et installation d'un fichier" l'exécution d'un programme quelconque. Ce programme peut, en particulier, contenir les transformations "fichier réel - syntaxe de transfert". Dans le même esprit, TFMM doit pouvoir être utilisé directement par des applications informatiques (toujours sous la responsabilité d'un utilisateur). On définit pour cela un protocole d'accès à TFMM. Cela ne pose pas de problèmes

puisque les primitives de service sont déjà formalisées. Après ces évolutions, il me semble que TFMM devient un outil complet et indispensable.

Mais, ce n'est que lorsqu'il sera très largement répandu et utilisé que l'on aura démontré que le système TFMM est intéressant. Pour contribuer à cela, deux types d'actions sont entreprises. D'une part le document norminatif a été confié à M. Stéfani pour qu'il le présente et le défende devant les instances de normalisation. Il est également communiqué à tous ceux qui le désire. D'autre part, le système TFMM a été réalisé sur deux systèmes d'exploitation : Unix et Multics. C'était tout à fait indispensable pour valider nos idées. Aujourd'hui, bien que l'implémentation ne soit pas totalement terminée, nous n'avons rencontré aucune incohérence, tout au plus deux ou trois petites erreurs. Le fait que cette implémentation est été achetée par plusieurs sociétés dont Cimsa, Bull , Primatel montre l'intérêt qui est porté à TFMM.

Ce projet a été enrichissant à plusieurs titres. Il m'a permis de me spécialiser dans un domaine de l'informatique en pleine expansion. Le fait qu'il débouche à la fois sur un document nominatif et sur une réalisation m'a donné une idée précise de la façon dont devait être mené un projet informatique à la fois réaliste et ambitieux. Pour terminer, je dirais que l'opportunité qu'il m'a donné de travailler avec des personnes d'horizons divers (Cnet, Bull, Cigref, BNI) m'a beaucoup apporté.

## REFERENCES

- [1] Recommandation X200. Modèle de référence pour l'interconnexion des systèmes ouverts pour les applications du CCITT. (Malaga - Torremolinos, 1984)
- [2] Recommandation X210. Conventions pour la définition des services de couches OSI. (Malaga - Torremolinos, 1984)
- [3] Recommandation X400. Systèmes de Messagerie : Modèle de systèmes et éléments de service. (Malaga - Torremolinos, 1984)
- [4] Recommandation X409. Systèmes de Messagerie : Syntaxe et notation de transfert de messages. (Malaga - Torremolinos, 1984)
- [5] Recommandation X410. Systèmes de Messagerie : Couche transfert de messages. (Malaga - Torremolinos, 1984)
- [6] Recommandation X420. Systèmes de Messagerie : Couche agent utilisateur de messagerie de personne à personne. (Malaga - Torremolinos, 1984)
- [7] Réseaux et télématique, Tomes 1 et 2. G.Pujolle, D.Seret, D.Dromard, E.Horlait.
- [8] Acte de Congrès, De Nouvelles Architectures pour les Communications. Impacts et enjeux de la normalisation. 28, 30 octobre 86, Paris. Eyrolles.
- [9] Open Systems Interconnection. New International Standards Architecture and Protocols for Distributed Information Systems. Proceedings of IEEE, Vol 71, No 12, December 1983
- [10] Description of Basic Reference Model for Open Systems Interconnection. ISO 7498.
- [11] Open Systems Interconnection : file transfer, acces and management. ISO/DP 8571/1-4.
- [12] Open Systems Interconnexion : specification of abstract syntax notation one (ANS.1). ISO/DIS 8824.
- [13] Open Systems Interconnexion : specification of basic encoding rules for abstract syntax notation one (ANS.1). ISO/DIS 8825.

- [14] Office Document Architecture. ECMA/TC29/83/56 ou ISO/TC96/SC18/WG3.

## **ANNEXE A**

Cette annexe contient un extrait du document final TFMM.

### **1 SERVICES FTUAL**

La couche FTUAL fournit aux utilisateurs (personnes, applications) les services de transfert de fichiers en mode messagerie.

On peut regrouper ces services en trois catégories (non indépendantes) : les services liés aux opérations, les services liés aux autorisations et enfin les services liés à l'UFS.

#### **1.1 SERVICES LIES AUX OPERATIONS**

Les services liés aux opérations sont fournis aux utilisateurs sous forme de primitives de service.

On présente d'abord les différentes primitives de service. Puis, on fournit des explications complémentaires sur les paramètres de ces dernières.

##### **1.1.1 Primitive de Demande d'Envoi de Fichier (DEF)**

L'utilisateur émet une primitive de Demande d'Envoi de Fichier pour demander l'envoi de la copie d'un fichier vers un ou plusieurs FTUAE impliquée(s) (cf Figure 1).

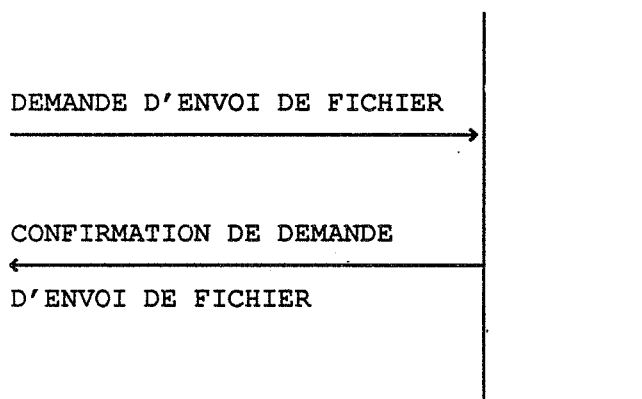


Figure 1

Les paramètres de la primitive de Demande d'Envoi de Fichier sont les suivants :

Nom du paramètre	Type
FTUAE initiatrice	M
FTUAE impliquée(s)	M
Demande de résultat d'envoi de fichier	M
Référence utilisateur	M
Commentaire	C
Nom du fichier sur le site initiateur	M
Nom d'autorisation d'arrivée	M
Nom du fichier sur le(les) site(s) impliqué(s)	M
Nature du fichier	M
Priorité de transfert	M

### 1.1.2 Primitive de Confirmation de Demande d'Envoi de Fichier (CDEF)

L'entité FTUAE répond à l'utilisateur ayant émis une primitive de Demande d'Envoi de Fichier par une primitive de Confirmation de Demande d'Envoi de Fichier (cf Figure 1).

Les paramètres de la primitive de Confirmation de Demande d'Envoi de Fichier sont les suivants :

Nom du paramètre	Type	Notes
------------------	------	-------

Référence utilisateur	M	
Date de demande	M	
Valeur de la confirmation	M	
Identificateur de demande	C	1
Dates associées au fichier copié	C	1
Raison de la confirmation négative	C	2

1. Ce paramètre ne figure que si la confirmation est positive.
2. Ce paramètre ne figure que si la confirmation est négative.

### 1.1.3 Primitive d'Indication de Signalisation d'Envoi de Fichier (ISEF)

L'entité FTUAE impliquée émet une primitive d'Indication de Signalisation d'Envoi de Fichier pour prévenir l'utilisateur qu'une opération d'envoi de fichier a eu lieu (cf Figure 2).

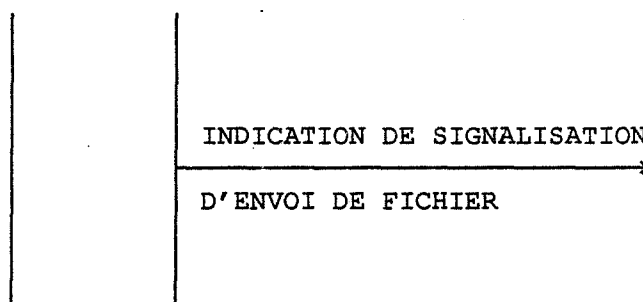


Figure 2

Les paramètres de la primitive d'Indication de Signalisation d'Envoi de Fichier sont les suivants :

Nom du paramètre	Type	Notes
FTUAE initiatrice	M	
FTUAE impliquée	M	
Référence utilisateur	M	
Identificateur de signalisation	M	
Date de traitement	M	



Date de demande	M	
Commentaire	C	
Nom du fichier sur le site initiateur	M	
Nom de l'autorisation d'arrivée	M	
Nom du fichier sur le site impliqué	M	
Priorité de transfert	M	
Nature du fichier	M	
Dates associées au fichier copié	C	
Valeur de la signalisation	M	
Renseignements d'installation	C	1
Raison de la signalisation négative	C	2

1. Ce paramètre ne figure que si la signalisation d'envoi de fichier est positive.
2. Ce paramètre ne figure que si la signalisation d'envoi de fichier est négative.

#### 1.1.4 Primitive d'Indication de Résultat d'Envoi de Fichier (IREF)

L'entité FTUAE initiatrice émet une primitive d'Indication de Résultat d'Envoi de Fichier pour prévenir l'utilisateur du résultat de l'envoi de fichier pour une des FTUAE impliquées (cf Figure 3).

L'utilisateur initiateur recevra autant de primitives d'Indication de Résultat d'Envoi de Fichier qu'il y avait de FTUAE impliquées dans la primitive de Demande d'Envoi de Fichier.

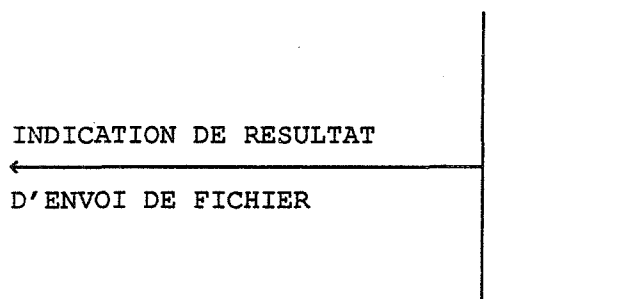


Figure 3

Les paramètres de la primitive d'Indication de Résultat d'Envoi de Fichier sont les suivants :

Nom du paramètre	Type	Notes
FTUAE initiatrice	M	
FTUAE impliquée	M	
Référence utilisateur	M	
Identificateur de demande	M	
Date du résultat	M	
Valeur du résultat	M	
Date de traitement	C	1
Renseignements d'installation	C	2
Raison du résultat négatif	C	3
Raison du résultat anormal	C	4
Date du résultat anormal	C	4

1. Ce paramètre ne figure que si le résultat d'envoi de fichier est positif ou négatif.
2. Ce paramètre ne figure que si le résultat d'envoi de fichier est positif.
3. Ce paramètre ne figure que si le résultat d'envoi de fichier est négatif.
4. Ce paramètre ne figure que si le résultat d'envoi de fichier est anormal.

#### 1.1.5 Primitive de Demande de Prise d'un Fichier (DPF)

L'utilisateur émet une primitive de Demande de Prise de Fichier pour demander l'installation par l'entité FTUAE d'une copie d'un fichier se trouvant sur un site distant (cf Figure 4).

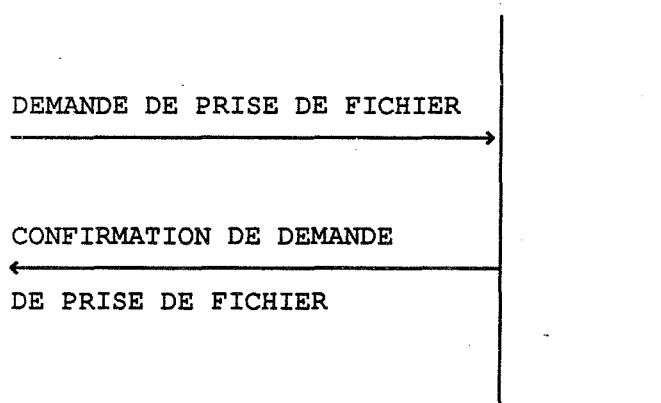


Figure 4

Les paramètres de la primitive de Demande de Prise de Fichier sont les suivants :

Nom du paramètre	Type
FTUAE initiatrice	M
FTUAE impliquée	M
Demande de résultat de prise de fichier	M
Référence utilisateur	M
Commentaire	C
Nom du fichier sur le site initiateur	M
Nom d'autorisation de départ	M
Nom du fichier sur le site impliqué	M
Nature du fichier	M
Priorité de transfert	M

#### 1.1.6 Primitive de Confirmation de Demande de Prise de Fichier (CDPF)

L'entité FTUAE répond à l'utilisateur ayant émis une primitive de Demande de Prise de Fichier par une primitive de Confirmation de Demande de Prise de Fichier (cf Figure 4).

Les paramètres de la primitive de Confirmation de Demande de Prise de Fichier sont les suivants :

Nom du paramètre	Type	Notes
------------------	------	-------

Référence utilisateur	M	
Date de demande	M	
Valeur de la confirmation	M	
Identificateur de demande	C	1
Raison de la confirmation négative	C	2

1. Ce paramètre ne figure que si la confirmation est positive.
2. Ce paramètre ne figure que si la confirmation est négative.

### 1.1.7 Primitive d'Indication de Signalisation de Prise de Fichier (ISPF)

L'entité FTUAE impliquée émet une primitive d'Indication de Signalisation de Prise de Fichier pour prévenir l'utilisateur qu'une opération de prise de fichier a eu lieu (cf Figure 5).

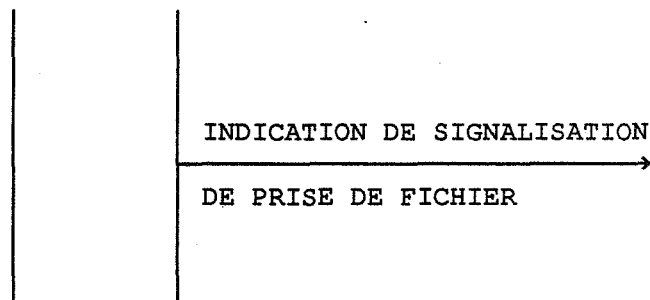


Figure 5

Les paramètres de la primitive d'Indication de Signalisation de Prise de Fichier sont les suivants :

Nom du paramètre	Type	Notes
FTUAE initiatrice	M	
FTUAE impliquée	M	
Référence utilisateur	M	
Identificateur de traitement	M	
Date de traitement	M	
Date de signalisation	M	

Commentaire	C	
Nom du fichier sur le site initiateur	M	
Nom de l'autorisation de départ	M	
Nom du fichier sur le site impliqué	M	
Priorité de transfert	M	
Nature du fichier	M	
Valeur de la signalisation	M	
Dates associées au fichier copié	C	1
Raison de la signalisation négative	C	2

1. Ce paramètre ne figure que si la signalisation de prise de fichier est positive.
2. Ce paramètre ne figure que si la signalisation de prise de fichier est négative.

#### 1.1.8 Primitive d'Indication de Résultat de Prise de Fichier (IRPF)

L'entité FTUAE initiatrice émet une primitive d'Indication de Résultat de Prise de Fichier pour prévenir l'utilisateur du résultat de la prise de fichier (cf Figure 6).

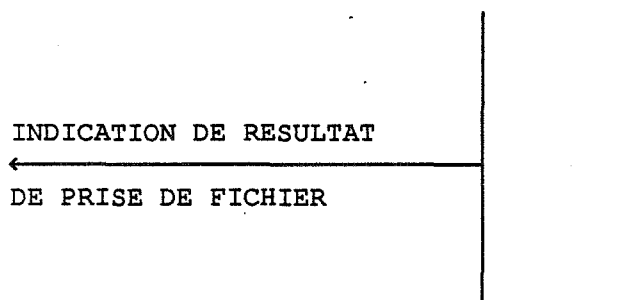


Figure 6

Les paramètres de la primitive d'Indication de Résultat de Prise de Fichier sont les suivants :

Nom du paramètre	Type	Notes
FTUAE initiatrice	M	
FTUAE impliquée	M	

Référence utilisateur	M	
Identificateur de demande	M	
Date du résultat	M	
Valeur du résultat	M	
Date de traitement	C	1
Dates associées au fichier copié	C	2
Renseignements d'installation	C	2
Raison de résultat négatif	C	3
Raison de résultat anormal	C	4
Date du résultat anormal	C	4

1. Ce paramètre ne figure que si le résultat de prise de fichier est positif ou négatif.
2. Ce paramètre ne figure que si le résultat de prise de fichier est positif.
3. Ce paramètre ne figure que si le résultat de prise de fichier est négatif.
4. Ce paramètre ne figure que si le résultat de prise de fichier est anormal.

#### 1.1.9 Primitive de Demande de Faisabilité d'Envoi (DFE)

L'utilisateur émet une primitive de Demande de Faisabilité d'Envoi pour savoir si un envoi de fichier aurait abouti dans les conditions actuelles (nom de FTUAE, désignation, autorisations, etc...) (cf Figure 7).

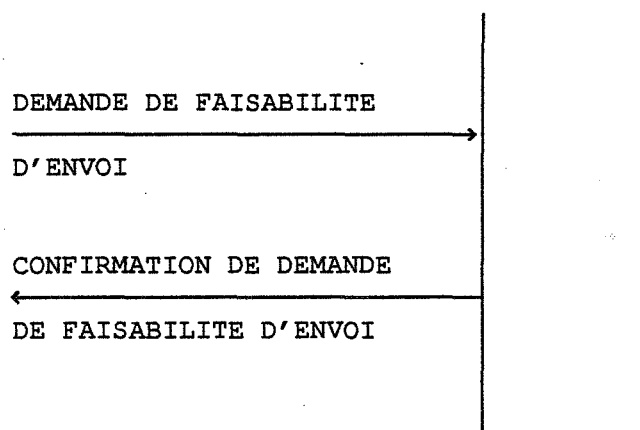


Figure 7

Les paramètres de la primitive de Demande de Faisabilité d'Envoi sont les suivants :

Nom du paramètre	Type
FTUAE initiatrice	M
FTUAE impliquées	M
Référence utilisateur	M
Commentaire	C
Nom du fichier sur le site initiateur	M
Nom d'autorisation d'arrivée	M
Nom du fichier sur le site impliqué	M
Nature du fichier	M
Priorité de transfert	M

#### 1.1.10 Primitive de Confirmation de Demande de Faisabilité d'Envoi (CDFE)

L'entité FTUAE répond par une primitive de Confirmation de Demande de Faisabilité d'Envoi à l'utilisateur ayant émis une primitive de Demande de Faisabilité d'Envoi (cf Figure 7).

Les paramètres de la primitive de Confirmation de Demande de Faisabilité d'Envoi sont les suivants :

Nom du paramètre	Type	Notes
Référence utilisateur	M	
Date de la demande	M	
Valeur de la confirmation	M	
Identificateur de demande	C	1
Dates associées au fichier copié	C	1
Taille du fichier	C	1
Raison de la confirmation négative	C	2

1. Ce paramètre ne figure que si la confirmation est positive.
2. Ce paramètre ne figure que si la confirmation est négative.

### 1.1.11 Primitive d'Indication de Résultat de Faisabilité d'Envoi (IRFE)

L'entité FTUAE initiatrice émet une primitive d'Indication de Résultat de Faisabilité d'Envoi pour mettre à la disposition de l'initiateur le résultat de la faisabilité d'envoi pour une des FTUAE impliquées (cf Figure 8). L'utilisateur initiateur recevra autant de primitives d'Indication de Résultat de Faisabilité d'Envoi qu'il y avait de FTUAE impliquées dans la primitive de Demande de Faisabilité d'Envoi.

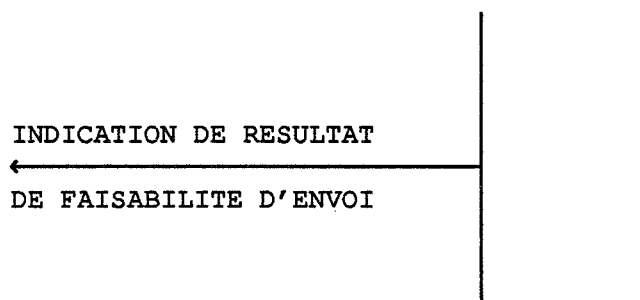


Figure 8

Les paramètres de la primitive d'Indication de Résultat de Faisabilité d'Envoi sont les suivants :

Nom du paramètre	Type	Notes
FTUAE initiatrice	M	
FTUAE impliquée	M	
Référence utilisateur	M	
Identificateur de demande	M	
Date du résultat	M	
Valeur du résultat	M	
Date de traitement	C	1
Possibilité d'installation	C	2
Raison du résultat négatif	C	3
Raison du résultat anormal	C	4
Date du résultat anormal	C	4

1. Ce paramètre ne figure que si le résultat de faisabilité d'envoi est positif ou négatif.
2. Ce paramètre ne figure que si le résultat de faisabilité d'envoi est positif.



3. Ce paramètre ne figure que si le résultat de faisabilité d'envoi est négatif.
4. Ce paramètre ne figure que si le résultat de faisabilité d'envoi est anormal.

### **1.1.12 Paramètres des primitives liées aux opérations**

#### **1.1.12.1 FTUAE initiatrice**

C'est l'entité FTUAE associée à l'utilisateur qui initialise une opération d'envoi de fichier, de prise de fichier ou de faisabilité d'envoi. Elle est désignée par un nom d'O/R.

#### **1.1.12.2 FTUAE impliquée(s)**

Ce sont le(s) FTUAE distant(s) impliquée(s) dans une opération d'envoi de fichier, de prise de fichier ou de faisabilité d'envoi. Elles sont désignées par des noms d'O/R.

#### **1.1.12.3 Demande de résultat d'envoi de fichier**

L'utilisateur indique s'il veut recevoir le(s) résultat(s) associé(s) à l'envoi de fichier.

#### **1.1.12.4 Demande de résultat de prise de fichier**

L'utilisateur indique s'il veut recevoir le résultat de la prise de fichier.

#### **1.1.12.5 Commentaire**

L'utilisateur initiateur en dispose pour commenter l'opération. Ce commentaire est mis, par l'entité FTUAE impliquée à la disposition de l'utilisateur auquel elle est associée. L'utilisation du commentaire est optionnelle.

#### **1.1.12.6 Nom du fichier sur le site initiateur**

Dans le cas d'un envoi de fichier, c'est le nom du fichier du site initiateur à copier et à envoyer. Dans le cas d'une prise de fichier, c'est le nom du fichier à installer sur le site initiateur. Dans le cas d'une faisabilité d'envoi, c'est le fichier faisant l'objet d'un essai d'envoi. Le fichier est désigné dans l'UFS, par une désignation absolue.

#### **1.1.12.7 Nom du fichier sur le(s) site(s) implique(s)**

Dans le cas d'un envoi de fichier, c'est le nom du fichier à installer sur le site impliqué. Dans le cas d'une prise de fichier, c'est le nom du fichier à copier sur le site impliqué. Dans le cas d'une faisabilité d'envoi, c'est le nom du fichier qui doit faire l'objet d'une vérification d'autorisation d'arrivée et d'un essai d'installation sur le site impliqué. Le fichier est désigné dans l'UFS, par une désignation

relative.

#### **1.1.12.8 Nature du fichier**

C'est la nature du fichier à transférer. Les cas prévus actuellement sont les natures texte et indéterminée. Dans le protocole P10, les syntaxes de transfert associées sont appelées texte-ligne et flot.

#### **1.1.12.9 Priorité de transfert**

C'est la priorité de transfert de niveau MTL associée à l'opération.

#### **1.1.12.10 Taille du fichier**

C'est la taille du fichier du site initiateur.

#### **1.1.12.11 Nom d'autorisation d'arrivée**

C'est le nom de l'autorisation d'arrivée associée à une opération d'envoi de fichier ou de faisabilité d'envoi.

#### **1.1.12.12 Nom d'autorisation de départ**

C'est le nom de l'autorisation de départ associée à une opération de prise de fichier.

#### **1.1.12.13 Référence utilisateur**

La référence utilisateur est fournie par l'utilisateur initiateur dans les paramètres des primitives de Demande. On la retrouve dans les primitives de Confirmation de Demande et dans les primitives d'Indication de Résultats. La référence utilisateur est également fournie à(aux) utilisateur(s) impliqué(s) dans les primitives d'Indication de Signalisation.

#### **1.1.12.14 Identificateur de demande d'opérations**

C'est l'identificateur fourni par l'entité FTUAE initiatrice. Il est associé à une primitive de Demande d'Envoi de Fichier avec demande de résultat, une Demande de Prise de Fichier avec demande de résultat ou une Demande de Faisabilité d'Envoi ; ces demandes ayant fait l'objet d'une confirmation positive.

Il n'est unique et significatif qu'au point d'accès où la demande a été émise. Il permet de mettre en correspondance une demande avec le(s) résultat(s) associé(s).

**1.1.12.15 Identificateur de signalisation**

C'est l'identificateur fourni par l'entité FTUAE impliquée. Il est associé à une primitive de Signalisation d'Envoi de Fichier ou de Signalisation de Prise de Fichier.

Il n'est unique et significatif qu'au point d'accès où la signalisation a été émise. Il permet d'identifier de façon non ambiguë toutes les signalisations.

**1.1.12.16 Date de demande**

C'est la date à laquelle l'utilisateur initiateur a demandé l'opération.

**1.1.12.17 Date de traitement**

C'est la date à laquelle l'entité FTUAE impliquée a traité l'opération.

**1.1.12.18 Date de résultat**

C'est la date la primitive d'Indication de Résultat de l'Envoi de Fichier, de Prise de Fichier ou de Faisabilité d'Envoi.

**1.1.12.19 Date du résultat anormal**

C'est la date à laquelle le résultat anormal a été détecté.

**1.1.12.20 Valeur de la confirmation**

Une confirmation peut être positive ou négative. Elle indique si l'entité FTUAE initiatrice accepte ou refuse une demande d'opération.

**1.1.12.21 Raison de la confirmation négative**

Ce sont principalement des erreurs d'ordre syntaxique dans les paramètres. Dans le cas d'opération d'envoi ou de faisabilité, elle peut indiquer que le fichier initiateur n'existe pas.

**1.1.12.22 Valeur de la signalisation**

Une signalisation peut être positive ou négative.

**1.1.12.23 Raison de la signalisation négative**

Ce sont les raisons pour lesquelles une opération d'envoi de fichier ou de prise de fichier a échoué sur le site impliqué.

Dans le cas d'un envoi, les raisons sont soit celles provenant de la vérification d'autorisation d'arrivée : nom d'autorisation d'arrivée incorrect, utilisateur initiateur non autorisé, fichier impliqué non autorisé; soit celles provenant de l'installation du fichier : problème de décodage, incompatibilité avec les droits de l'utilisateur impliqué, problèmes divers à la création du fichier impliqué.

Dans le cas de prise de fichier, les raisons sont soit celles provenant de la vérification d'autorisation de départ : nom d'autorisation de départ incorrect, utilisateur initiateur non autorisé, fichier impliqué non autorisé; soit celles provenant de la copie de fichier impliqué : incompatibilité avec les droits de l'utilisateur impliqué, problèmes divers à la copie du fichier impliqué; soit celle provenant du codage du fichier : problème de codage. fichier impliqué non autorisé.

#### **1.1.12.24 Valeur du résultat**

Le résultat peut être positif, négatif ou anormal.

#### **1.1.12.25 Raison du résultat négatif**

Ce sont les raisons d'un résultat négatif d'opération.

Dans le cas d'un envoi, les raisons sont les mêmes que pour la signalisation négative.

Dans le cas de prise de fichier, les raisons sont celles de signalisation négative auxquelles s'ajoutent celles provenant du décodage du fichier, celles provenant de l'installation du fichier : incompatibilité avec les droits de l'utilisateur initiateur, problèmes divers à la création du fichier sur les site initiateur.

Dans le cas d'une faisabilité, les raisons sont celles provenant de la vérification d'autorisation d'arrivée : nom d'autorisation d'arrivée incorrect, utilisateur initiateur non autorisé,

#### **1.1.12.26 Raison du résultat anormal**

C'est une anomalie de niveau MTL ou un problème de niveau FTUAL.

L'entité FTUAE dispose de cet élément de protocole P10-Problème pour indiquer qu'un problème (protocole incorrect, ressources insuffisantes, autres) est survenu à la suite de la réception d'un élément de protocole.

#### **1.1.12.27 Dates associées au fichier copié**

Ce sont les dates associées au fichier copié : date de création, date de dernier accès et date de dernière modification. Le fichier copié est le fichier du site initiateur dans le cas d'un envoi de fichier et le fichier du site impliqué dans le cas d'une prise de fichier. Les dates sont fournies à condition que le SGF réel les gère.

#### **1.1.12.28 Possibilités d'installation**

Indique, dans le cas de résultat positif de faisabilité d'envoi, si l'installation du fichier sur le site impliqué serait possible. Les conditions étudiées sont les droits locaux de l'utilisateur impliqué, la taille du fichier, l'existence d'un fichier, etc... Ce paramètre figure dans la mesure où l'entité FTUAE a pu le vérifier.

#### **1.1.12.29 Renseignements d'installation**

Ce sont les renseignements sur l'installation du fichier sur le site impliqué dans le cas d'une opération d'envoi de fichier ou sur le site initiateur dans le cas d'une opération de prise de fichier. Ils indiquent s'il y a eu création de répertoires, ou, création du fichier sous un autre nom dans le cas où il existait déjà un fichier de ce nom.

## 1.2 SERVICES LIES AUX AUTORISATIONS

Les services liés aux autorisations sont fournis aux utilisateurs sous forme de primitives de service.

### 1.2.1 Primitive de Demande de Don d'Autorisation d'Arrivée (DDAA)

L'utilisateur émet une primitive de Demande de Don d'Autorisation d'Arrivée pour donner une autorisation d'arrivée (cf Figure 9). Une autorisation d'arrivée est valable jusqu'à sa suppression par l'utilisateur.

L'entité FTUAE impliquée n'acceptera une opération d'envoi de fichier ou de faisabilité d'envoi que s'il existe une autorisation d'arrivée du nom indiqué et si les différents paramètres de l'autorisation sont compatibles avec ceux de l'opération.

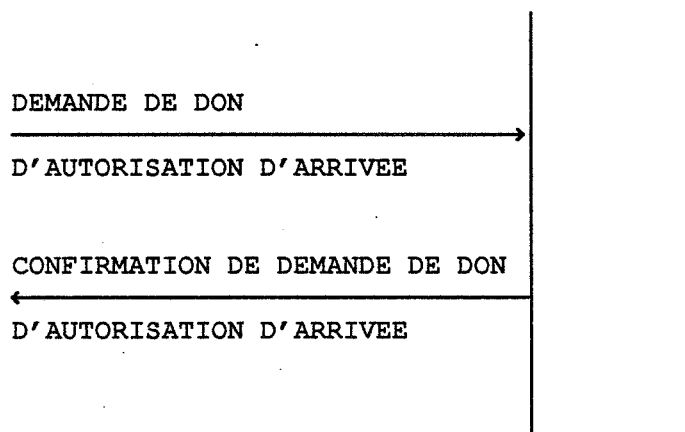


Figure 9

Les paramètres de la primitive de Demande de Don d'Autorisation d'Arrivée sont les suivants :

Nom du paramètre	Type	Notes
Nom d'autorisation d'arrivée	M	
Zone visible	M	
Utilisateur(s) autorisé(s)	M	
Fichier(s) autorisé(s)	C	1
Répertoire(s) autorisé(s)	C	1

1. Il faut au moins l'un des deux paramètres.

### 1.2.2 Primitive de Confirmation de Demande de Don d'Autorisation d'Arrivée (CDDAA)

L'entité FTUAE répond à l'utilisateur ayant émis une primitive de Demande de Don d'Autorisation d'Arrivée par une primitive de Confirmation de Demande de Don d'Autorisation d'Arrivée (cf Figure 9).

Les paramètres de la primitive de Confirmation de Demande de Don d'Autorisation d'Arrivée sont les suivants :

Nom du paramètre	Type	Notes
Valeur de la confirmation	M	
Raison de la confirmation négative	C	1

1. Ce paramètre ne figure que si la confirmation est négative.

### 1.2.3 Primitive de Demande de Don d'Autorisation de Départ (DDAD)

L'utilisateur émet une primitive de Demande de Don d'Autorisation de Départ pour donner une autorisation de départ (cf Figure 10). Une autorisation de départ est valable jusqu'à sa suppression par l'utilisateur.

L'entité FTUAE impliquée n'acceptera une opération de prise de fichier que s'il existe une autorisation de départ du nom indiqué et si les différents paramètres de l'autorisation de départ sont compatibles avec ceux de l'opération.

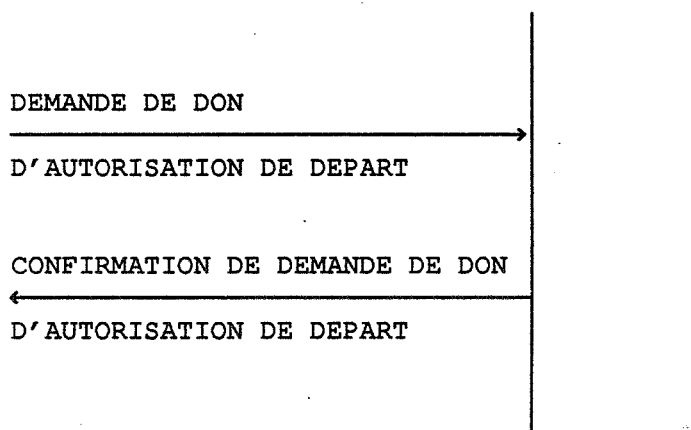


Figure 10

Les paramètres de la primitive de Demande de Don d'Autorisation de Départ sont les suivants :

Nom du paramètre	Type	Notes
------------------	------	-------

Nom d'autorisation d'arrivée	M	
Zone visible	M	
Utilisateur(s) autorisé(s)	M	
Fichier(s) autorisé(s)	C	1
Répertoire(s) autorisé(s)	C	1

1. Il faut au moins l'un des deux paramètres.

#### 1.2.4 Primitive de Confirmation de Demande de Don d'Autorisation de Départ (CDDAD)

L'entité FTUAE répond à l'utilisateur ayant émis une primitive de Demande de Don d'Autorisation de Départ par une primitive de Confirmation de Demande de Don d'Autorisation de Départ (cf Figure 10).

Les paramètres de la primitive de Confirmation de Demande de Don d'Autorisation de Départ sont les suivants :

Nom du paramètre	Type	Notes
Valeur de la confirmation	M	
Raison de la confirmation négative	C	1

1. Ce paramètre ne figure que si la confirmation est négative.

#### 1.2.5 Primitive de Demande de Suppression d'Autorisation d'Arrivée (DSAA)

L'utilisateur émet une primitive de Demande de Suppression d'Autorisation d'Arrivée pour supprimer une autorisation d'arrivée (cf Figure 11).



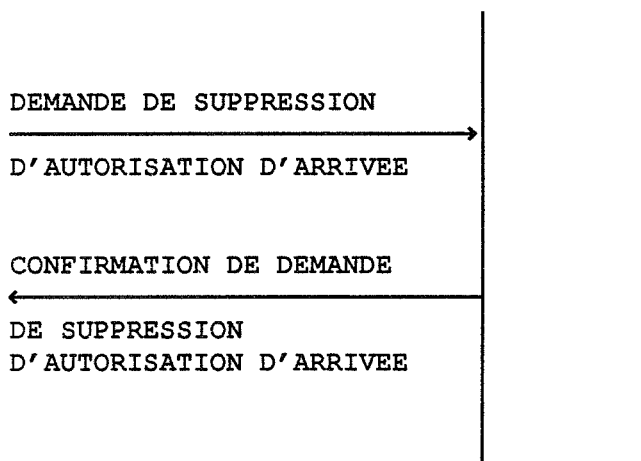


Figure 11

Les paramètres de la primitive de Demande de Suppression d'Autorisation d'Arrivée sont les suivants :

Nom du paramètre	Type	Notes
Nom de l'autorisation d'arrivée	M	

#### 1.2.6 Primitive de Confirmation de Demande de Suppression d'Autorisation d'Arrivée (CDSAA)

L'entité FTUAE répond à l'utilisateur ayant émis une primitive de Demande de Suppression d'Autorisation d'Arrivée par une primitive de Confirmation de Demande de Suppression d'Autorisation d'Arrivée (cf Figure 11).

Les paramètres de la primitive de Confirmation de Demande de Suppression d'Autorisation d'Arrivée sont les suivants :

Nom du paramètre	Type	Notes
Valeur de la confirmation	M	
Raison de la confirmation négative	C	1

1. Ce paramètre ne figure que si la confirmation est négative.

### 1.2.7 Primitive de Demande de Suppression d'une Autorisation de Départ (DSAD)

L'utilisateur émet une primitive de Demande de Suppression d'Autorisation de Départ pour supprimer une autorisation de départ (cf Figure 12).

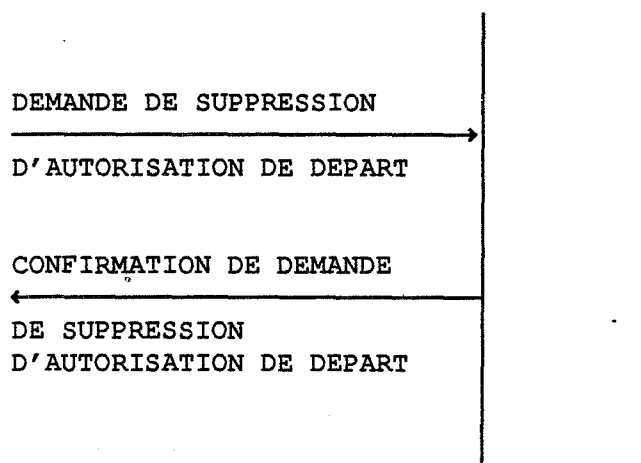


Figure 12

Les paramètres de la primitive de Demande de Suppression d'Autorisation de Départ sont les suivants :

Nom du paramètre	Type	Notes
Nom d'autorisation de départ	M	

### 1.2.8 Primitive de Confirmation de Demande de Suppression d'Autorisation de Départ (CDSAD)

L'entité FTUAE répond à l'utilisateur ayant émis une primitive de Demande de Suppression d'Autorisation de Départ par une primitive de Confirmation de Demande de Suppression d'Autorisation de Départ (cf Figure 12).

Les paramètres de la primitive de Confirmation de Demande de Suppression d'Autorisation de Départ sont les suivants :

Nom du paramètre	Type	Notes
Valeur de la confirmation	M	
Raison de la confirmation négative	C	1

1. Ce paramètre ne figure que si la confirmation est négative.

## **1.2.9 Paramètres des primitives liées aux autorisations**

### **1.2.9.1 Nom d'une autorisation d'arrivée**

C'est le nom associé à une autorisation de d'arrivée. A un moment donné, il ne peut y avoir plusieurs autorisations d'arrivée désignées par le même nom. L'utilisateur initiateur d'une opération d'envoi de fichier ou de faisabilité d'envoi doit fournir le nom de l'autorisation d'arrivée correspondante.

### **1.2.9.2 Nom d'une autorisation de départ**

C'est le nom associé à une autorisation de départ. A un moment donné il ne peut y avoir plusieurs autorisations de départ désignées par le même nom. L'utilisateur initiateur d'une opération de prise de fichier doit fournir le nom de l'autorisation de départ correspondante.

### **1.2.9.3 Utilisateurs autorisés**

Ce sont les utilisateurs distants associés à une autorisation de départ ou d'arrivée. Ils sont désignés par des noms d'O/R.

### **1.2.9.4 Zone visible**

C'est la zone visible associée à une autorisation de départ ou d'arrivée. C'est le répertoire à partir duquel les fichiers et répertoires autorisés sont désignés.

### **1.2.9.5 Fichiers autorisés**

Ce sont les fichiers autorisés associés à une autorisation de départ ou d'arrivée. Les fichiers sont désignés en UFS par une désignation relative par rapport à la zone visible.

### **1.2.9.6 Répertoires autorisés**

Ce sont les répertoires autorisés associés à une autorisation de départ ou d'arrivée. Sous ces répertoires tous les fichiers sont autorisés. Les répertoires autorisés sont désignés en UFS par une désignation relative par rapport à la zone visible.

### **1.2.9.7 Valeur de la confirmation**

Une confirmation peut être positive ou négative.

### **1.2.9.8 Raison de la confirmation négative**

Donne la raison pour laquelle la confirmation est négative.

### **1.3 SERVICES LIES AU SYSTEME DE FICHIER UNIFORMISE**

Ce service consiste à offrir à l'utilisateur et aux entités FTUAE distantes une vision UFS du SGF courant. Il est à la charge des implémenteurs d'établir la correspondance UFS-SGF.

#### **1.3.1 Nom simple UFS**

Un nom simple UFS est le nom associé à un fichier ou à un répertoire de l'arborescence UFS.

#### **1.3.2 Désignation UFS**

##### **1.3.2.1 Désignation UFS absolue**

Une désignation absolue UFS est la suite des noms simples qui commence à la racine de l'arborescence et qui conduit au fichier ou répertoire auquel elle est associée.

##### **1.3.2.2 Désignation UFS relative**

Une désignation relative UFS est la suite des noms simples qui commence à un répertoire quelconque et qui conduit au fichier ou répertoire auquel elle est associée.

#### **1.3.3 Nature de fichier**

Les services offerts par l'entité FTUAE sont le codage et le décodage d'un fichier SGF suivant sa nature. Les natures de fichier actuellement gérées par les FTUAE sont le flot et le texte.

## 2 PROTOCOLE P10

Nous présentons dans cette section le protocole P10 utilisé par les entités FTUAE.

### 2.1 IDENTIFICATEUR DE LANCEMENT

L'identificateur de lancement est associé à une opération. Il est généré par l'entité FTUAE initiatrice et il accompagne tous les éléments de protocole P10 associés à cette opération.

### 2.2 CHAÎNE-TFMM

Ce sont les caractères utilisés pour les noms simples de fichier, les noms d'autorisations, le commentaire et la référence utilisateur. C'est tous les caractères IA5 exceptés les caractères de contrôle.

### 2.3 DEFINITION DU PROTOCOLE P10

P10 DEFINITIONS ::=

BEGIN

```
FTPDU ::= SEQUENCE {  
    id-lancement [0] IMPLICIT Chaîne-IA5,  
    [1] CHOICE {  
        [0] IMPLICIT Lancement,  
        [1] IMPLICIT Réponse,  
        [2] IMPLICIT Problème }};
```

```
Lancement ::= SEQUENCE {  
    date-demande [0] IMPLICIT Date-Heure,  
    [1] CHOICE {  
        [0] IMPLICIT Lancement-Envoi,  
        [1] IMPLICIT Lancement-Prise,  
        [2] IMPLICIT Lancement-Faisabilité }};
```

```
Réponse ::= SEQUENCE {  
    date-traitement [0] IMPLICIT Date-Heure,  
    [1] CHOICE {  
        [0] Réponse-Envoi,  
        [1] Réponse-Prise,  
        [2] Réponse-Faisabilité }}}
```

```
Problème ::= SEQUENCE {  
    date-problème [0] IMPLICIT Date-Heure,  
    type-problème [1] IMPLICIT INTEGER {  
        protocole-incorrupt (0),  
        ressources-insuffisantes (1),  
        autres (2) }}
```

```
Lancement-Envoi ::= SEQUENCE {  
    paramètres-envoi [0] IMPLICIT Paramètres,  
    fichier-envoyé [1] Fichier,  
    dates-fichier [3] IMPLICIT Dates,  
    demande-résultat [4] IMPLICIT BOOLEAN }
```

```
Lancement-Prise ::= SEQUENCE {  
    paramètres-prise [0] IMPLICIT Paramètres,  
    demande-résultat [1] IMPLICIT BOOLEAN }
```

```
Lancement-Faisabilité ::= SEQUENCE {  
    paramètres-faisabilité [0] IMPLICIT Paramètres,  
    taille-fichier [1] IMPLICIT INTEGER }
```

```
Réponse-Envoi ::= CHOICE {  
    [0] IMPLICIT Réponse-Envoi-Positive,  
    [1] IMPLICIT Réponse-Envoi-Négative }
```

```
Réponse-Envoi-Positive ::= Renseignements-Installation
```

Réponse-Envoi-Négative ::= INTEGER {  
    nom-autorisation-arrivée-incorrupt (0),  
    utilisateur-non-autorisé (1),  
    fichier-impliqué-non-autorisé (2),  
    incompatibilité-droits-utilisateur-impliqué (3),  
    problème-décodage (4),  
    problème-divers-installation (5)}

Réponse-Prise ::= CHOICE {  
    [0] IMPLICIT Réponse-Prise-Positive,  
    [1] IMPLICIT Réponse-Prise-Négative }

Réponse-Prise-Positive ::= SEQUENCE {  
    nature [0] IMPLICIT Nature-Fichier,  
    nom-initiateur [1] IMPLICIT Désignation-UFS-Absolue,  
    fichier-pris [2] Fichier,  
    dates-fichier [3] IMPLICIT Dates }

Réponse-Prise-Négative ::= INTEGER {  
    nom-autorisation-depart-incorrupt (0),  
    utilisateur-non-autorisé (1),  
    fichier-impliqué-non-autorisé (2),  
    incompatibilité-droits-utilisateur-impliqué (3),  
    fichier-impliqué-inexistant (4),  
    problème-codage (5)}

Réponse-Faisabilité ::= CHOICE {  
    [0] IMPLICIT Réponse-Faisabilité-Positive,  
    [1] IMPLICIT Réponse-Faisabilité-Négative }

Réponse-Faisabilité-Positive ::= Possibilité-Installation

Réponse-Faisabilité-Négative ::= INTEGER {  
    nom-autorisation-arrivée-incorrupt (0),  
    utilisateur-non-autorisé (1),  
    fichier-impliqué-non-autorisé (2)}

Paramètres ::= SEQUENCE {  
    ref-utilisateur [0] IMPLICIT Référence-Utilisateur,  
    com [1] IMPLICIT Commentaire OPTIONAL,  
    nature [2] IMPLICIT Nature-Fichier,  
    nom-initiateur [3] IMPLICIT Désignation-UFS-Absolue,  
    nom-autorisation [4] IMPLICIT Nom-Autorisation,  
    nom-impliqué [5] IMPLICIT Désignation-UFS-Relative }

Renseignements-Installation ::= INTEGER {  
    création-alias (0),  
    création-repertoires (1),  
    rien-à-signaliser (2) }

Possibilité-Installation ::= INTEGER {  
    ok (0),  
    non-vérifiable (1)  
    incompatibilité-droits-utilisateur-impliqué (2),  
    fichier-déjà-existant (3),  
    espace-disque (4),  
    autres (5)}

Date-Heure ::= Heure-UTC

Référence-Utilisateur ::= Chaîne-TFMM

Commentaire ::= Chaîne-TFMM



Nature-Fichier ::= INTEGER {  
    texte (0),  
    indéterminée (1) }

Nom-Autorisation ::= Chaîne-TFMM

Désignation-UFS-Absolue ::= SEQUENCE of Nom-Simple

Désignation-UFS-Relative ::= SEQUENCE of Nom-Simple

Nom-Simple ::= Chaîne-TFMM

Chaîne-TFMM ::= [ APPLICATION 1] IMPLICIT Chaîne-IA5

Dates ::= SEQUENCE {  
    date-cr ation [1] IMPLICIT Date-Heure OPTIONAL,  
    date-acc s [2] IMPLICIT Date-Heure OPTIONAL,  
    date-maj [3] IMPLICIT Date-Heure OPTIONAL }

Fichier ::= CHOICE {  
    [0] IMPLICIT Texte-Ligne,  
    [1] IMPLICIT Flot }

Texte-Ligne ::= SEQUENCE of Ligne

Ligne ::= Chaîne-IA5

Flot ::= Octet-String

END

### 3 ACTIONS SUR RECEPTION DE PRIMITIVES

#### 3.1 ACTIONS SUR RECEPTION DE PRIMITIVES UTILISATEURS

Les primitives utilisateurs sont les suivantes :

- Demande d'Envoi de Fichier (DEF)
- Demande de Prise de Fichier (DPF)
- Demande de Faisabilité d'Envoi (DFE)
- Demande de Don d'Autorisation d'Arrivée (DDAA)
- Demande de Don d'Autorisation de Départ (DDAD)
- Demande de Suppression d'Autorisation d'Arrivée (DSAA)
- Demande de Suppression d'Autorisation de Départ (DSAD)

##### 3.1.1 Réception d'une Primitive de Demande d'Envoi de Fichier

L'entité FTUAE vérifie les différents paramètres. Elle prend une copie du fichier à envoyer. Elle le code suivant la nature. Elle génère un identificateur de lancement. Elle fabrique un élément de protocole P10 (Lancement, Lancement-Envoi).

Elle émet auprès de la couche MTL une primitive de Demande de Dépôt dont les paramètres sont les suivants. Les destinataires sont les entités FTUAE impliquées. Le contenu est le P10 fabriqué. Le type de contenu est P10. La priorité est la priorité de transfert. L'avis de non remise est supprimé sauf dans le cas de demande de résultat. Les autres paramètres ne sont pas utilisés.

L'entité FTUAE répond à l'utilisateur par une primitive de Confirmation de Demande d'Envoi de Fichier. La confirmation est positive ou négative.

Dans le cas négatif, l'entité FTUAE refuse la primitive de Demande d'Envoi de Fichier. Elle fournit à l'utilisateur les raisons de ce refus. Il n'y a eu pas d'envoi de P10.

Dans le cas positif, l'entité FTUAE accepte la primitive de Demande d'Envoi de Fichier. S'il y a demande de résultat, elle génère un identificateur de demande.

##### 3.1.2 Réception d'une Primitive de Demande de Prise de Fichier

L'entité FTUAE vérifie les différents paramètres. Elle génère un identificateur de lancement. Elle fabrique un élément de protocole P10 (Lancement, Lancement-Prise).

Elle émet auprès de la couche MTL une primitive de Demande de Dépôt dont les paramètres sont les suivants. Le destinataire est l'entité FTUAE impliquée. Le contenu est le P10 fabriqué. Le type de contenu est P10. La priorité est la priorité de transfert. L'avis de non remise est supprimée sauf dans le cas de demande de résultat. Les autres paramètres ne sont pas utilisés.

L'entité FTUAE répond à l'utilisateur par une primitive de Confirmation de Demande de Prise de Fichier. La confirmation est positive ou négative.

Dans le cas négatif, l'entité FTUAE refuse la primitive de Demande de Prise de Fichier. Elle fournit à l'utilisateur les raisons de ce refus. Il n'y a eu pas d'envoi de P10.

Dans le cas positif, l'entité FTUAE accepte la primitive de Demande de Faisabilité d'Envoi. S'il y a demande de résultat, elle génère un identificateur de demande.

### **3.1.3 Réception d'une Primitive de Demande de Faisabilité d'Envoi**

L'entité FTUAE vérifie les différents paramètres. Il génère un identificateur de lancement. Elle fabrique un élément de protocole P10 (Lancement, Lancement-Faisa).

Elle émet auprès de la couche MTL une primitive de Demande de Dépôt dont les paramètres sont les suivants. Les destinataires sont les entités FTUAE impliquées. Le contenu est le P10 fabriqué. Le type de contenu est P10. La priorité est la priorité de transfert. Les autres paramètres ne sont pas utilisés.

L'entité FTUAE répond à l'utilisateur par une primitive de Confirmation de Demande de Faisabilité d'Envoi. La confirmation est positive ou négative.

Dans le cas négatif, l'entité FTUAE refuse la primitive de Demande de Faisabilité d'Envoi. Elle fournit à l'utilisateur les raisons de ce refus. Il n'y a eu pas de fabrication et d'envoi de P10.

Dans le cas positif, l'entité FTUAE accepte primitive de Demande de Faisabilité d'Envoi. Elle génère un identificateur de demande.

### **3.1.4 Réception d'une Primitive de Demande de Don d'Autorisation d'Arrivée**

L'entité FTUAE répond à l'utilisateur par une primitive de Confirmation de Demande de Don d'Autorisation d'Arrivée. La confirmation est négative s'il existe déjà une autorisation d'arrivée portant le même nom. Les autres raisons de refus sont des problèmes d'ordre syntaxique sur les différents paramètres de la demande. Il n'y a pas d'envoi de protocole P10.

### **3.1.5 Réception d'une Primitive de Demande de Don d'Autorisation de Départ**

L'entité FTUAE répond à l'utilisateur par une primitive de Confirmation de Demande de Don d'Autorisation de Départ.

La confirmation est négative s'il existe déjà une autorisation de départ portant le même nom. Les autres raisons de refus sont des problèmes d'ordre syntaxique sur les différents paramètres de la demande. Il n'y a pas d'envoi de protocole P10.

### **3.1.6 Réception d'une Primitive de Demande de Suppression d'Autorisation d'Arrivée**

L'entité FTUAE répond à l'utilisateur par une primitive de Confirmation de Demande de Suppression d'Autorisation d'Arrivée. La confirmation est négative s'il n'existe pas d'autorisation portant ce nom. Il n'y a pas d'envoi de protocole P10.

### **3.1.7 Réception d'une Primitive de Demande de Suppression d'Autorisation de Départ**

L'entité FTUAE répond à l'utilisateur par une primitive de Confirmation de Demande de Suppression d'Autorisation de Départ. La confirmation est négative s'il n'existe pas d'autorisation portant ce nom. Il n'y a pas d'envoi de protocole P10.

### 3.2 ACTIONS SUR RECEPTION DE PRIMITIVES MTL

Les primitives MTL que peut recevoir l'entité FTUAE sont :

- Indication de Remise (IR)
- Confirmation de Demande de Dépôt (CDD)
- Indication d'Avis (IA)

#### 3.2.1 Réception d'une Primitive d'Indication de Remise

Une Indication de Remise contient du P10. Suivant la valeur du P10, l'entité FTUAE effectue les actions suivantes :

##### 3.2.1.1 Indication de Remise contenant un élément de protocole P10 Lancement-Envoi

L'entité FTUAE, qui est donc l'entité FTUAE impliquée, vérifie l'autorisation d'arrivée. Il en déduit la zone visible. Il decode le fichier suivant sa nature et l'installe. Il émet auprès de l'utilisateur impliqué une primitive d'Indication de Signalisation d'Envoi de Fichier.

S'il y a demande de résultat, l'entité FTUAE génère un élément de protocole P10 (Réponse, Réponse-Envoi) contenant l'identificateur de lancement. C'est une Réponse-Envoi-Positive, si l'entité FTUAE a pu installer le fichier, une Réponse-Envoi-Négative sinon.

L'entité FTUAE émet auprès de la couche MTL une primitive de Demande de Dépôt avec les paramètres suivants. Le contenu est le P10 fabriqué. Le Type de Contenu est P10. Le destinataire est l'expéditeur de la primitive d'Indication de Remise. La priorité est la priorité de la primitive d'Indication de Remise. L'avis de non remise est supprimé. Les autres paramètres ne sont pas utilisés.

##### 3.2.1.2 Indication de Remise contenant un élément de protocole P10 Lancement-Prise

L'entité FTUAE, qui est donc l'entité FTUAE impliquée, vérifie l'autorisation de départ. Il en déduit la zone visible. Il prend une copie du fichier et le code suivant sa nature. Il émet auprès de l'utilisateur une primitive d'Indication de Signalisation de Prise de Fichier.

S'il y a demande de résultat ou si la prise de fichier a réussi, l'entité FTUAE génère un élément de protocole P10 (Réponse, Réponse-Prise) contenant l'identificateur de lancement. L'entité FTUAE émet auprès de la couche MTL une primitive de Demande de Dépôt avec les paramètres suivants. Le contenu est le P10. Le type de contenu est P10. Le destinataire est l'expéditeur de la primitive d'Indication de Remise. La priorité est la priorité de la primitive d'Indication de

Remise. L'avis de non remise est supprimé. Les autres paramètres ne sont pas utilisés.

### **3.2.1.3 Indication de Remise contenant un élément de protocole P10 Lancement-Faisa**

L'entité FTUAE, qui est donc l'entité FTUAE impliquée, vérifie l'autorisation d'arrivée. Si l'autorisation est vérifiée, l'entité FTUAE étudie les possibilités d'installation suivant la taille du fichier et les droits locaux de l'utilisateur. Il peut exister des sites où l'entité FTUAE n'a aucun moyen d'étudier les possibilités d'installation.

L'entité FTUAE génère un élément de protocole P10 (Réponse, Réponse-Faisa) contenant l'identificateur de lancement. C'est une Réponse-Faisa-Positive si l'autorisation a été vérifiée, une Réponse-Faisa-Négative sinon.

L'entité FTUAE émet auprès de la couche MTL une primitive de Demande de Dépôt avec les paramètres suivants. Le contenu est le P10. Le destinataire est l'expéditeur de la primitive d'Indication de Remise. La priorité est la priorité de la primitive d'Indication de Remise. L'avis de non remise est supprimée. Les autres paramètres ne sont pas utilisés.

### **3.2.1.4 Indication de Remise contenant un élément de protocole P10 Réponse-Envoi**

L'entité FTUAE émet auprès de l'utilisateur une primitive d'Indication de Résultat d'Envoi de Fichier.

### **3.2.1.5 Indication de Remise contenant un élément de protocole P10 Réponse-Prise**

Si c'est une Réponse-Prise-Positive, l'entité FTUAE décode le fichier suivant sa nature et l'installe. S'il y a demande de résultat, l'entité FTUAE émet une primitive d'Indication de Résultat de Prise de Fichier positive ou négative suivant le résultat du décodage et l'installation de fichier.

Si c'est une Réponse-Prise-Négative, l'entité FTUAE émet auprès de l'utilisateur une primitive d'Indication de Résultat de Prise de Fichier.

### **3.2.1.6 Indication de Remise contenant un élément de protocole P10 Réponse-Faisabilité**

L'entité FTUAE émet auprès de l'utilisateur une primitive d'Indication de Résultat de Faisabilité d'Envoi.

### **3.2.1.7 Indication de Remise contenant un élément de protocole P10 Problème**

Si le problème suit un Lancement d'Envoi avec demande résultat, de Prise avec demande de résultat ou de Faisabilité d'Envoi, l'entité FTUAE émet, auprès de l'utilisateur, une primitive d'Indication de Résultat d'Envoi de Fichier, de Prise de Fichier ou de Faisabilité d'Envoi.

Le résultat est de type anormal. Les raisons du résultat anormal et la date du résultat anormal sont déduites du P10.

### **3.2.2 Réception d'une Primitive de Confirmation de Demande de Dépôt**

Si la primitive de Confirmation de Demande de Dépôt est négative et si elle suit une Demande de Dépôt contenant un Lancement d'Envoi avec demande de résultat, de Prise avec demande de résultat ou de Faisabilité, l'entité FTUAE émet, auprès de l'utilisateur, une primitive d'Indication de Résultat d'Envoi de Fichier, de Prise de Fichier ou de Faisabilité d'Envoi.

Le résultat est de type anormal. Les raisons du résultat anormal et la date du résultat anormal sont déduites de la Confirmation de Demande de Dépôt. Dans les autres cas, l'entité FTUAE ne fait rien.

### **3.2.3 Réception d'une Primitive d'Indication d'Avis**

La primitive d'Indication d'Avis est toujours de non remise. Elle suit une Demande de Dépôt contenant un Lancement d'Envoi avec demande de résultat, de Prise avec demande de résultat ou de Faisabilité. L'entité FTUAE émet, auprès de l'utilisateur, une primitive d'Indication de Résultat d'Envoi de Fichier, de Prise de Fichier ou de Faisabilité d'Envoi.

Le résultat est de type anormal. Les raisons du résultat anormal et la date du résultat anormal sont déduites de l'indication d'avis.





## **ANNEXE B**

Cette annexe contient la présentation générale de GTF.

### **1 INTRODUCTION**

L'application GTF réalise les services de transfert de fichiers (TFMM). Elle fait partie de COSAC V5.

L'application GTF est constituée d'un ensemble de modules et d'objets.

On présente ici rapidement ces différents modules et objets. Pour plus de précisions, se rapporter aux documents suivants :

- GTF : Les Objets
- GTF : Le Module DO
- GTF : Le Module CB
- GTF : Le Module MJA
- GTF : Le Module GTF-GM
- GTF : Le Module GESGTF

### **2 LES OBJETS COSAC**

L'application GTF utilise certains objets de l'application COSAC : PARGM, GESUA, FAS, FAE, STOCK, REQUA (cf documents COSAC V5).

### **3 LES OBJETS GTF**

#### **3.1 LES OBJETS PARGTF**

Il y en a un par utilisateur. Un objet PARGTF contient différents paramètres GTF propres à l'utilisateur.

### **3.2 LES OBJETS JOUR**

Il y en a un par utilisateur. Un objet JOUR est constitué d'objets TRANS qui contiennent des renseignements sur les opérations lancées par l'utilisateur et qui ne sont pas encore terminées.

### **3.3 LES OBJETS BADR (BOITE A DEMANDE-RESULTAT)**

Il y en a un par utilisateur. Un objet BADR est constitué d'objets DEMRES qui correspondent à des demandes des opérations (envoi, prise et faisabilité) lancés par l'utilisateur, et aux résultats associés.

### **3.4 LES OBJETS BAS (BOITE A SIGNALISATION)**

Il y en a un par utilisateur. Un objet BAS est constitué d'objets SIG qui correspondent à des signalisations reçues par l'utilisateur.

### **3.5 LES OBJETS LAA (LISTE D'AUTORISATIONS D'ARRIVEE)**

Il y en a un par utilisateur. Un objet LAA est constitué d'objets ARR. Un objet ARR correspond à une autorisation d'arrivée donnée par l'utilisateur.

### **3.6 LES OBJETS LAD (LISTE D'AUTORISATIONS DE DEPART)**

Il y en a un par utilisateur. Un objet LAD est constitué d'objets DEP. Un objet DEP correspond à une autorisation de départ donnée par l'utilisateur.

## 4 L'ARBORESCENCE GTFOBJ

L'ensemble des objets GTF est organisé en arborescence sous le répertoire GTFOBJ.

Il est installé en même temps que l'ensemble du logiciel COSAC. C'est par défaut "cosac/gtfobj". On trouve son nom dans l'objet PARGM.

### 4.1 LE REPERTOIRE TEMPGTF

Sous le répertoire GTFOBJ, on trouve obligatoirement le répertoire TEMPGTF. Son nom est "tempgtf".

Il contient des fichiers temporaires utilisés par les différents modules de l'application GTF.

### 4.2 LE REPERTOIRE DATAGTF

Sous le répertoire GTFOBJ, on trouve obligatoirement le répertoire DATAGTF. Son nom est "datagtf".

Il contient plusieurs fichiers de données de l'application GTF : fichier de messages d'erreurs, etc...

### 4.3 LE REPERTOIRE PAR

Sous le répertoire GTFOBJ, on trouve obligatoirement le répertoire PAR. Son nom est "par".

Il contient pour chaque utilisateur un fichier de nom uti \* qui correspond à l'objet PARGTF.

### 4.4 LE REPERTOIRE JOURNAUX

Sous le répertoire GTFOBJ, on trouve obligatoirement le répertoire JOURNAUX. Son nom est "journaux".

Il contient pour chaque utilisateur un répertoire de nom uti \* qui correspond à l'objet JOUR.

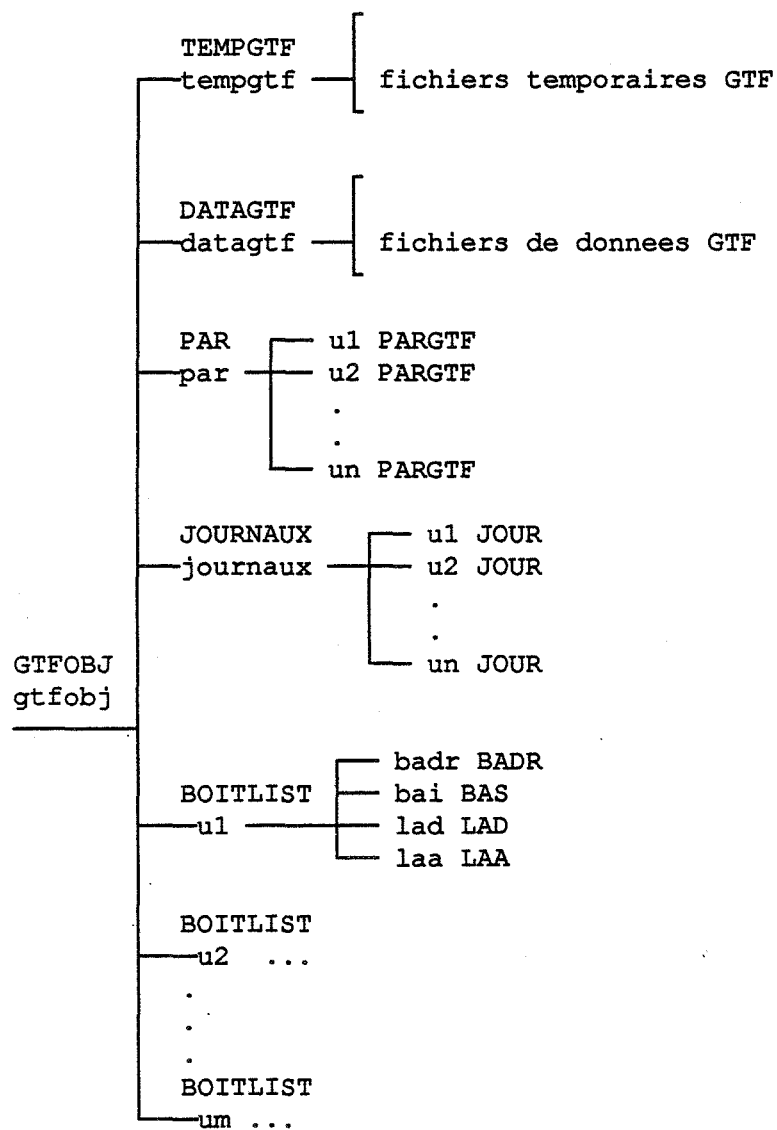
---

\* uti : nom de login de l'utilisateur

#### 4.5 LES REPERTOIRES BOITLIST

On définit pour chaque utilisateur un répertoire BOITLIST sous lequel on trouve 4 répertoires représentant les boites et les listes. Leurs noms sont " badr ", " bai ", " laa " et " lad ".

Le nom du répertoire BOITLIST peut être indiqué par l'utilisateur, on le trouve dans l'objet PARGTF associé à l'utilisateur. C'est par défaut GTFOBJ/uti \*.



u1,...,un : noms de login des utilisateurs

Figure 1 : Arborescence GTF OBJ

## 5 LE MODULE DE DEMANDE D'OPERATIONS : DO

Ce module correspond au traitement des primitives de Demande d'Envoi de Fichier, de Prise de Fichier et de Faisabilité d'Envoi. Le module DO est composé de 3 sous-modules DOENVOI, DOPRISE, DOFAISA qui sont activables par l'utilisateur.

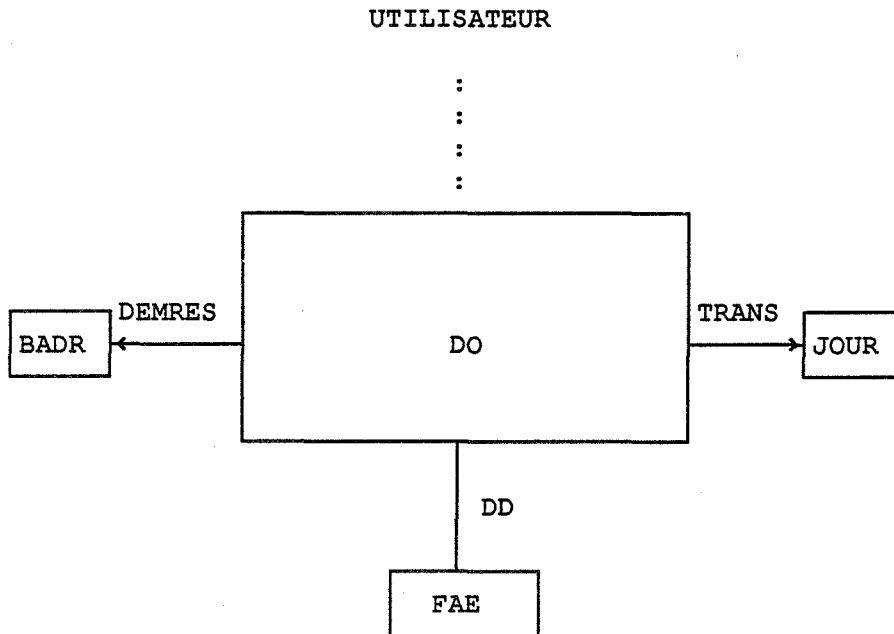


Figure 2 : Fonctionnement de DO

Ces 3 sous-modules commencent par la saisie et la vérification des paramètres fournis par l'utilisateur. Parmi ces paramètres, la référence utilisateur (idU) que l'on retrouvera par la suite dans les objets DEMRES et SIG associés à la demande.

Si le module DO détecte une erreur, il rend un compte-rendu négatif.

Sinon, le module DO enchaîne sur les différentes actions associées aux opérations

Il fabrique et code le P10. Le P10 contient en particulier l'Identificateur de Lancement (idlanc). Cet identificateur sera présent dans le P10 de Réponse ou de Problème.

Il dépose auprès de GM une primitive de Demande de Dépôt (DD) contenant le P10. Dans les paramètres de la DD, il fournit entre autres un identificateur d'UA-Contenu (idUA). On retrouvera cet identificateur dans la primitive de Confirmation de Demande de Dépôt (CD) et la primitive d'Indication d'Avis de Remise (IA) qui suivent la DD.

Il crée un ou plusieurs objets DEMRES dans le cas d'envoi avec demande de résultat, de prise avec demande de résultat et de faisabilité.

Un objet DEMRES contient les différents paramètres de la demande et de la confirmation de demande positive.

Un identificateur de Demande (iddem) est rendu à l'utilisateur. Cet identificateur est unique par rapport à toutes les demandes jamais effectuées par l'utilisateur. Cet identificateur est présent ainsi que l'identificateur utilisateur (idU) dans l'objet DEMRES.

Le module DO crée un objet TRANS. Cet objet contient entre autres l'identificateur de Demande (iddem), l'identificateur de Lancement (idlanc), l'identificateur d'UA-Contenu (idUA) et l'identificateur d'événement Demande de Dépôt (idDD).



## 6 MODULE DE CONSULTATIONS DES BOITES : CB

Le module CB permet à l'utilisateur de lire et détruire les objets (DEMRES, SIG) contenus dans les objets (BADR, BAS).

Le module CB est composé de 2 sous-modules CBADR et CBAS qui sont activables par l'utilisateur.

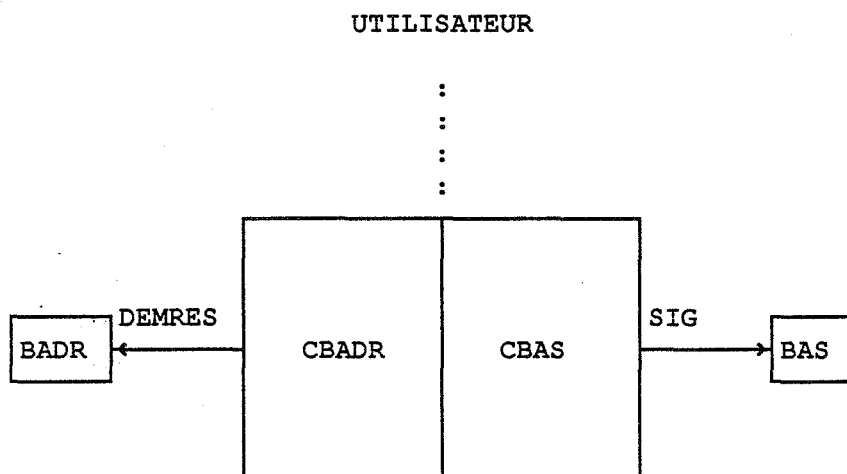


Figure 3 : Fonctionnement de CB

### 6.1 CONSULTATION DE LA BOITE A DEMANDE-RESULTAT : CBADR

On rappelle qu'un objet BADR contient un ensemble d'objets DEMRES. Un objet DEMRES contient une demande d'opération lancée par l'utilisateur et les résultats associés reçus.

CBADR commence par fournir à l'utilisateur un sommaire de tous les objets DEMRES présents au moment de l'appel.

Ce sommaire est de la forme :

numéro	iddem	idU
--------	-------	-----

L'utilisateur peut à l'aide de ce numéro référencer un objet DEMRES pour le lire et/ou le détruire.

La destruction d'un objet DEMRES n'est effective qu'à la sortie de CBADR.

L'utilisateur peut détruire un objet DEMRES pour lequel tous les résultats ne sont pas parvenus. Si, par la suite, les résultats arrivent, ils ne seront pas conservés.

Si pendant une activation de CBADR, de nouveaux objets DEMRES sont créés par le module DO, ils ne pourront être consultés qu'à la prochaine activation de CBADR.

Par contre, si pendant l'activation de CBADR, des résultats associés à un objet DEMRES parviennent, ils sont consultables immédiatement.

## 6.2 CONSULTATION DE LA BOITE A SIGNALISATION : CBAS

On rappelle qu'un objet BAS contient un ensemble d'objets SIG. Un objet SIG correspond à une primitive d'Indication de Signalisation reçue par l'utilisateur.

CBAS commence par fournir à l'utilisateur un sommaire de tous les objets SIG présents dans l'objet BAS au moment de l'appel. Ce sommaire est de la forme :

numéro	idsig	idU
--------	-------	-----

L'utilisateur peut à l'aide de ce numéro référencer un objet SIG pour le lire et/ou le détruire.

La destruction d'un objet SIG n'est effective qu'à la sortie de CBAS.

Si pendant une activation de CBAS, de nouveaux objets SIG sont créés par le module GTF-GM, ils ne pourront être consultés qu'à la prochaine activation de CBAS.

## 7 MODULE DE MISE A JOUR DES AUTORISATIONS : MJA

Le module MJA permet à l'utilisateur de créer, lire, modifier et détruire les objets (ARR, DEP) contenus dans les objets (LAA, LAD).

Le module MJA est composé de 2 sous-modules MJAA et MJAD activables par l'utilisateur.

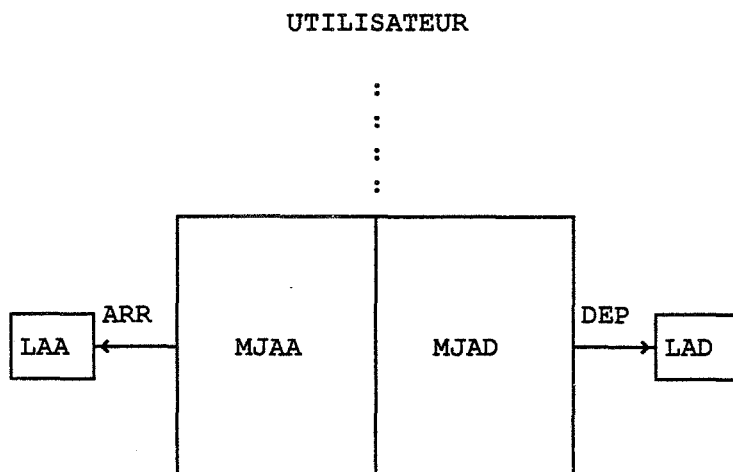


Figure 4 : Fonctionnement de MJA

### 7.1 MISE A JOUR DES AUTORISATION D'ARRIVEE : MJAA

On rappelle qu'un objet LAA contient un ensemble d'objets ARR. Un objet ARR correspond à une autorisation d'arrivée donné à un utilisateur ou un groupe d'utilisateur distants.

MJAA commence par fournir à l'utilisateur un sommaire de tous les objets ARR présents.

Ce sommaire est de la forme :

numéro	nom d'autorisation d'arrivée
--------	------------------------------

L'utilisateur peut lire, détruire, ajouter ou modifier des objets ARR. Ces modifications sont immédiatement répercuter sur le sommaire que l'utilisateur peut demander à consulter à tout moment.

**ATTENTION :** Les modifications sur l'objet LAA ne sont effectives qu'à la sortie de MJAA. Si pendant l'activation de MJAA, le module GTF-GM accède à l'objet LAA pour vérifier une autorisation, il consultera l'ancienne version.

## **7.2 MISE A JOUR DES AUTORISATIONS DE DEPART :MJAD**

On rappelle qu'un objet LAD contient un ensemble d'objets DEP. Un objet DEP correspond à une autorisation de départ donné à un utilisateur ou un groupe d'utilisateur distant.

Le fonctionnement de MJAD est analogue à celui de MJAA.

## 8 MODULE GTF-GM

Le module GTF-GM traite les primitives MTA émises par GM . Il est commun à tous les utilisateurs. Il prend successivement l'identification du FTUA destinataire de la primitive MTA et travaille sur les objets associés à l'utilisateur. Il est activé automatiquement sous la responsabilité de l'administrateur GTF.

Il extrait de la FAS les primitives. Une primitive est soit une Indication de Remise (IR), soit une Confirmation de Demande de Dépot (CD), soit une Indication d'Avis de Remise (IA).

Suivant les cas, il modifie l'objet DEMRES ou crée l'objet SIG, consulte les objets LAA ou LAD, modifie l'objet TRANS, génère des OPDU de Réponse ou de Problème et dépose une DD dans la FAE.

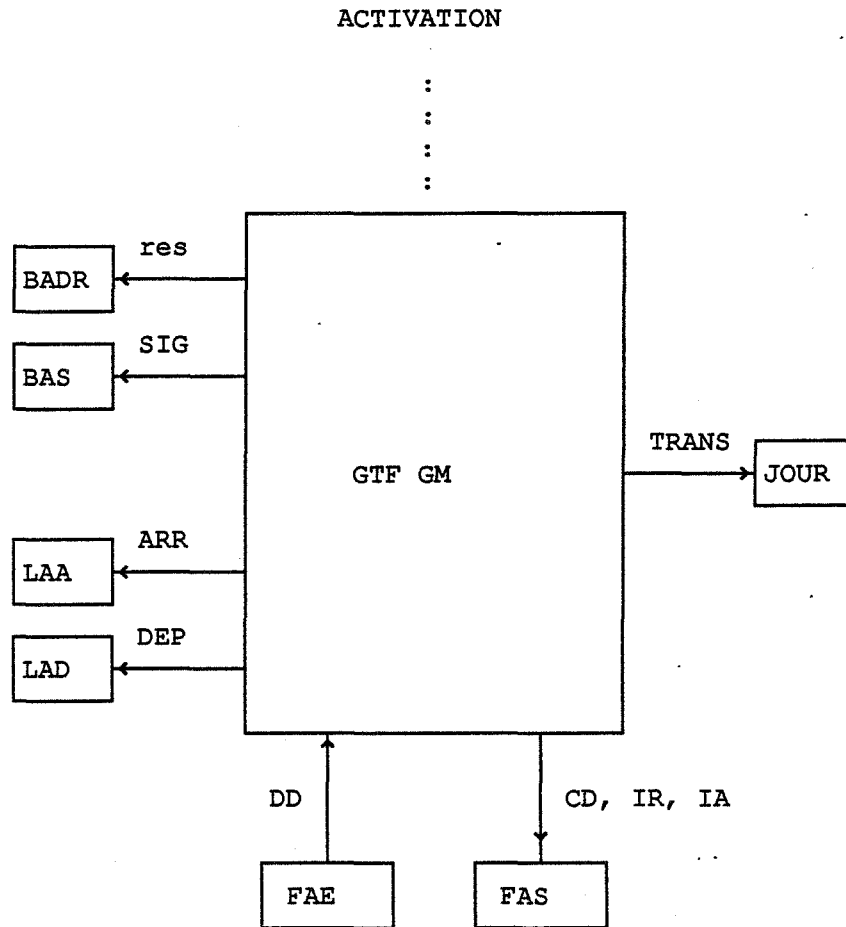


Figure 5 : Fonctionnement de GTF GM

### 8.1 ACTIONS SUR RECEPTION D'UNE PRIMITIVE CD

Si la Confirmation de Demande de Dépôt suit une Demande de Dépôt contenant un Lancement d'Envoi avec de résultat, de Prise avec demande de résultat ou Faisabilité, il existe un objet TRANS correspondant. GTF-GM le retrouve à l'aide de l'identificateur d'idUA-Contenu (idUA).

Si la CD est positive, GTF-GM rajoute l'identificateur d'événement de Demande de Dépôt (idDD) dans l'objet TRANS. Si la CD est négative, GTF-GM détruit l'objet TRANS.

Si la CD est négative, il rajoute les résultats dans l'objet(s) DEMRES. Il retrouve l'objet DEMRES grâce à l'identificateur de demande présent dans l'objet TRANS.

### 8.2 ACTIONS SUR RECEPTION D'UNE PRIMITIVE IA

Dans le module DO, on demande à recevoir uniquement les avis de non remise. Dans le module GTF-GM, on demande à ne pas recevoir d'avis. Une IA suit donc obligatoirement un DD contenant un P10 de Lancement ( idlanc) et elle est forcément négative. Elle concerne un ou plusieurs destinataires.

GTF-GM met à jour l'objet TRANS pour les destinataires concernés. Il le retrouve à l'aide de l'identificateur d'UA-Contenu (idUA) ou de l'identificateur de d'événement de Demande de Dépôt (iDD).

GTF-GM rajoute les résultats dans l'objet DEMRES. Il le retrouve à l'aide de l'identificateur de demande (iddem) présent dans l'objet TRANS.

### 8.3 ACTIONS SUR RECEPTION D'UNE PRIMITIVE IR

Une Indication de Remise contient un élément de protocole P10. GTF-GM commence par décoder le P10. Ce dernier est mis dans une structure pascal de type wftpdu (cf p10typ.p).

Cette structure établit des limitations sur certains sous-éléments. Par exemple sur la taille des identificateurs, du commentaire, etc... Mais ces limitations ne portent pas sur la taille du fichier transféré. Ces limitations sont en correspondance avec les informations contenus dans les objets boîtes et les listes.

Si GTF-GM détecte, au cours du décodage des erreurs, il génère un P10 de Problème qu'il transmet à GM.

Sinon il enchaîne suivant le type du P10.

### 8.3.1 Cas P10 de Lancement

Si le P10 est Lancement-Envoi, GTF-GM recherche dans l'objet LAA une autorisation d'arrivée ARR. Il en déduit la désignation absolu du fichier à installer. Il installe le fichier.

Si le P10 est Lancement-Prise, GTF-GM recherche dans l'objet LAD une autorisation de départ DEP. Il en déduit la désignation absolu du fichier à prendre. Il copie le fichier et le code suivant sa structure logique.

Si le P10 est Lancement-Faisa, GTF-GM recherche dans l'objet LAA une autorisation d'arrivée ARR. Il en déduit la désignation absolu du fichier qu'il devrait installer. Il étudie les conditions d'installation.

Suivant le résultat de l'opération, GTF-GM génère un P10 Réponse. Il code le P10. Il effectue une DD auprès de GM.

### 8.3.2 Cas P10 Réponse

Dans le cas d'envoi de fichier avec demande de résultat, de prise de fichier avec demande de résultat et de faisabilité d'envoi, il existe un objet TRANS. GTF-GM le retrouve grâce à l'identificateur de lancement. On trouve dans l'objet TRANS l'identificateur de demande associé à l'objet DEMRES.

Dans le cas d'une Réponse-Envoi, GTF-GM ajoute le résultat dans l'objet DEMRES associé.

Dans le cas d'une Réponse-Faisa, GTF-GM ajoute le résultat dans l'objet DEMRES associé.

Si le cas d'une Réponse-Prise-Positive, GTF-GM décode le fichier selon sa structure logique. Il l'installe. Si l'utilisateur a demandé à recevoir un résultat, GTF-GM ajoute le résultat dans l'objet DEMRES associé.

Si le cas d'une Réponse-Prise-Négative, GTF-GM ajoute le résultat dans l'objet DEMRES associé.

### 8.3.3 Cas P10 Problème

Dans le cas d'envoi de fichier avec demande de résultat, de prise de fichier avec demande de résultat et de faisabilité d'envoi, il existe un objet TRANS. GTF-GM le retrouve grâce à l'identificateur de lancement. On trouve dans l'objet TRANS l'identificateur de demande. GTF-GM ajoute le résultat dans l'objet DEMRES associé.



## **9 MODULE DE GESTION DE GTF : GESGTF**

Il permet à l'administrateur GTF (ADGTF) de gérer l'ensemble de l'application GTF.



